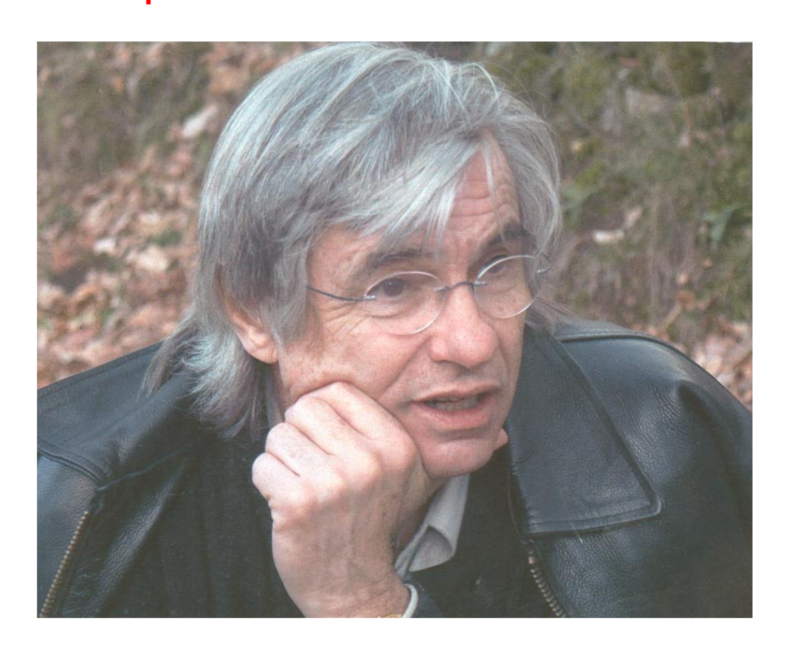
ACHILLE MONEYBACK DÉCOUVRE L'INFORMATIQUE



Savoir sans Frontières

Association Loi de 1901
Villa Jean-Christophe, 206 Chemin de la Montagnère, 84120 France

http://www.savoir-sans-frontieres.com



Jean-Pierre Petit, Président de l'Association : Ancien Directeur de Recherche au Cnrs, astrophysicien, créateur d'un genre nouveau : la Bande Dessinée Scientifique. En 2005 il a décidé de mettre ses ouvrages au nombre d'une vingtaine, dans le domaine public en les rendant téléchargeables gratuitement à partir de son site. Il a également créé l'association Savoir sans Frontières qui s'est donnée pour but de distribuer gratuitement le savoir, y compris le savoir scientifique et technique à travers le monde. L'association, qui fonctionne grâce à des dons, rétribue des traducteurs à hauteur de 150 euros (en 2006) en prenant à sa charge les frais d'encaissement bancaire. De nombreux traducteurs accroissent chaque jour le nombre d'albums traduits (en 2005 en 18 langues, dont le Laotien et le Rwandais).

Le présent fichier pdf peut être librement dupliqué et reproduit, en tout ou en partie, utilisé par les enseignants dans leurs cours à conditions que ces opérations ne se prêtent pas à des activités lucratives. Il peut être mis dans les bibliothèques municipales, scolaires et universitaires, soit sous forme imprimée, soit dans des réseaux de type Intranet.

L'auteur a entrepris de compléter cette collection par des albums plus simples d'abord (niveau 12 ans). Egalement en cours d'élaboration : des albums « parlants » pour analphabètes et « bilingues » pour apprendre des langues à partir de sa langue d'origine.

L'association recherche sans cesse de nouveaux traducteurs vers des langues qui doivent être leur langue maternelle, possédant les compétences techniques qui les rendent aptes à produire de bonne traductions des albums abordés.

Les dons (chèque libellés à l'ordre de Savoir sans Frontières) sont également les bienvenus. Les ressources de l'association sont en 2006 principalement affectées aux nouvelles traductions.

AVERTISSEMENT DE L'AUTEUR (1999)

Voici deux petits livres, mis bout à bout, publiés il y a seize ans déjà. Les éditeurs ont disparu depuis longtemps, de même que les livres.

De nos jours, on ne programme plus, ou presque. On laisse ça aux "pros", ou on consomme du logiciel. Mais, en retrouvant ces ouvrages dans mon grenier je me suis dit qu'après tout ils constituaient une petite initiation à l'informatique assez sympa. C'est du "vieux BASIC" avec "des numéros de lignes". Il aurait fallu, pour le moins, réécrire ces programmes en VISUAL BASIC, mais j'avoue avoir manqué de temps pour le faire. Même chose pour le logiciel PANGRAPHE de La CAO sans peine. Si un lecteur pouvait faire ce travail, je serais ravi de mentionner son nom et même de faire figurer sa photo dans la prochaine version du cd. Ça, plus un dessin original et un diplôme de membre bienfaiteur du club Lanturlu.

Jean Pierre Petit

GLOSSAIRE INFORMATIQUE

3	INFORMATIQUE	: Technique qui utilise l'ordinateur.
5	ORDINATEUR	: Machine programmable de multiples façons.
6	LOGICIEL	: Ensemble de programmes disponibles.
6	DISQUE LOURD	: Mémoire externe à très forte capacité.
8	MICRO-ORDINATEUR	: Ordinateur dont le "cerveau" est un microprocesseur.
8	MICROPROCESSEUR	: Cerveau, ou plutôt cervelet d'un ordinateur.
9	UNITÉ CENTRALE	: autre dénomination du "cervelet".
11	DISKETTE	: mémoire externe sous forme d'un disque souple, bon marché.
11	ECRAN VIDÉO	: Ecran TV pour afficher textes, chiffres, graphiques.
12	MEMOIRE	: Dans l'ordinateur, peut stocker des nombres, des chaines de caractères, des instructions composant des programmes.
13	ADRESSE	: Système de repérage des contenus.
14	REGISTRE	: Element, case mémoire, contenant une DONNÉE : Nombre ou chaine.
14	CHAINE DE CARACTÈRE	: Suite quelconque de signes.
14	DONNÉE	: Information (Numérique ou Chaine) susceptible d'être introduite dans l'ordinateur avant ou pendant l'exécution d'un programme.
14	INSTRUCTION	: Opération élementaire codée, mémorisable.
14	LANGAGE	: Ensemble d'instructions (qui sont les "mots" de ce langage).
14	SYNTAXE	: règles d'utilisation des instructions.
15	MESSAGE D'ERREUR	: Sanctionne une erreur (en général de syntaxe).
14	ÉCRITURE EN MÉMOIRE	: Introduction de qqch dans la mémoire.
14	LECTURE	: Interroger la mémoire, lire un registre, afficher son contenu.
17	ROM	: Architecture de l'ordinateur, ineffaçable, où l'utilisateur n'a pas accès.
17	RAM	: Zone mémoire où on peut ECRIRE (données ou programme).
19	PROGRAMME	: Suite ordonnée d'instructions, présentées en LIGNES-PROGRAMMES.
22	LANCER UN PROGRAMME	: Déclencher son exécution.
21	BASIC	: Un langage particulier des ordinateurs.
27	LISTER UN PROGRAMME	: Afficher ce programme, ligne par ligne, sur l'écran.

30	ENTRER DES DONNÉES : Alimenter les registres de l'ordinateur.
34	SAUVER UN PROGRAMME : Recopier un programme, à partir d'une mémoire centrale (volatile) sur une mémoire externe (non volatile), par exemple diskette.
34	MÉMOIRE VOLATILE : Contenu détruit en mettant le système hors tension.
34	CHARGER UN PROGRAMME : Recopier un programme de la mémoire externe vers la m. centrale.
36	BOUCLE : Structure d'une tâche répétitive dans l'ordinateur.
38	INCRÉMENTATION : augmentation du contenu d'un registre numérique.
38	INCRÉMENT : la quantité dont on INCRÉMENTE le registre-compteur.
38	DÉCRÉMENTER, DÉCRÉMENT : opération inverse.
39	INSTRUCTION DE BRANCHEMENT : Envoit vers une autre ligne-programme.
39	BOUCLE SANS FIN : structure où rien n'a été prévu pour sortir de boucle.
39	BRANCHEMENT INCONDITIONNEL : S'effectue à l'aide d'un GOTO
40	BRANCHEMENT CONDITIONNEL : Assorti d'un Test : IF ..... THEN....
42	BOUCLE DE TEMPORISATION : boucle "vide", qui maintient l'affichage sur écran.
51	OPTIONS : Différentes sections de programme vers lesquelles on peut s'orienter.
52	MENU : Structure de Programme permettant de multiples options.
53	SOUS-PROGRAMME : Section de programme qui peut être, dans le programme principal, exécutée "toutes affaires cessantes", ceci étant suivi d'une reprise du cours normal.
57	TRAITEMENT DE TEXTE : Manipulations de chaines de caractères.
59	SOUS-CHAINE : Un fragment d'une chaine de caractères.
60	VARIABLE : Registre dont le contenu peut varier en cours d'exécution.
61	VARIABLE INDICÉE : Registre à contenu variable, repéré à l'aide d'un indice.
65	FICHIER : Ensemble de registres repérés avec le (ou les) même indice.
67	RÉSERVATION DE PLACES MÉMOIRE : Affectation d'un secteur de la mémoire pour y loger un ou des fichiers, à un ou plusieurs indices.
74	FLAG : Registre où on note, d'une manière quelconque, si une opération a été effectuée ou non.


ma bichette
oui, mon ami, vous me l'avez déja dit cent fois
c'est l'informatisation ou le dépôt de bilan
Mais comme personne dans cette maison n'a été capable de me dire ce qu'était l'informatique, je me contenterai de bien penser à vous
Au SICOB, s'il vous plait
allons, viens, Dagobert !
Voyons, le petit Larousse INFORMATIQUE : technique de traitement automatique de l'information
Mon Dieu je sens que cela va être dur


VOUS ETES ICI
Courage...
PAR LÀ
S'il vous plait, où est-ce qu'on peut voir des ordinateurs ?
Ca dépend. Que voulez vous voir ? Des minis ? des micros ?
Euh... des moyens...
PIEGEAC 2000
Voyons... troisième niveau ... ..porte G ... stands huit à onze...
ALOUETTE PLACART
DALLAS INSTRUMENT
le pigeon-type
un petit conseil en passant, monsieur ?

Euh, Moneyback, de la société Achille Moneyback

Clavier, de la SYSTEX

Je dirige une petite entreprise, et nous avions pensé que, pour les stocks...

Hmm... je vois... relance des impayés... fichier conjoncturel. Monsieur Moneyback je vais être très direct avec vous. Savez vous ce qu'est un ordinateur ?

Euh... pas vraiment..

Très bien... Quelle importance, finalement. Vous êtes venu me voir et c'est l'essentiel. Et vous êtes venu À TEMPS! En tant que leader responsable, vous savez que, demain, pour TOUTES les entreprises, c'est.... .... l'informatisation ou..

Le dépôt de bilan ?..

Mais alors... un ordinateur, c'est quoi ?..


Je schématise. Vous voyez ce rond. Cela sera pour nous le système ENTRÉE-SORTIE que constitue l'ordinateur.
mais, l'entrée-sortie c'est QUOI ?...
Entrée, traitement, sortie. Ce sont les mots-clefs. Vous n'avez pas besoin d'en savoir plus !...
ENTRÉE
SORTIE
TRAITEMENT
Ah !..
.... notre nouveau PIÈGEAC 2000, avec son puissant logiciel polyvalent, entièrement conversationnel, avec saisie directe assistée et comptabilisation en temps réel, accès multiporte, disc mégaoctets en ligne, s'occuppe du reste...
Mais, son prix ?
Pour une configuration utilisable avec deux disques lourds pour commencer, en comptant la maintenance de logiciel, cela ferait...
6


Zut, mes piles sont à plat !
Ne pourrait-on pas utiliser votre merveilleuse machine ... votre PIEGEAC ?..
mais... bien sûr
Zut ! la tuile...
Bon, c'est simple, je charge l'operating system par CC300/A ce qui libère mon WORK SPACE
Ah, zut, il est en BATCH ! J'ai oublié de faire TTX c'est à dire une demande d'interruption prioritaire
Vous comprenez ?
Euh... oui !


l'informatique...
je n'y compendrai jamais rien
LE FORTRAN DANS LA JOIE
PRATIQUE DU 6502
LE PASCAL SANS PEINE
JE PARLE LE COBOL
Ouh là là ....
Ben dis donc...
SBOUNG!
!?
B
Excusez moi, messieurs, mais qu'est-ce que cet appareil fait au SICOB ?
C'est un microordinateur
Un ordinateur? on dirait une machine à sous!
Mais non, tout y est, l'UNITÉ CENTRALE, le MICROPROCESSEUR, les MÉMOIRES.
mais... je ne vois qu'un clavier et un écran de Télévision

Où est... le reste ?

Tout est là

Tenez, voila l'unité centrale

Ce sont ces petites choses qu'on trouve dans les jouets ?

Pas seulement. Un ordinateur peut faire des tas de choses très variées, vous savez...

Je passe une DISKETTE

?

Ça par exemple c'est un programme qu'a fait un copain du CNRS. Ça calcule des champs magnétiques.

CHAMP BØ=1000 GAUSS
DEUX BARRES A+/-2CM

Ah bon, ça fait aussi du calcul scientifique !

mais... pas seulement. Tenez, avec cet autre programme je transforme l'ordinateur en synthétiseur musical.

mais... la musique est sur ce... disque ?

Oui, mais pas comme vous croyez. Ça n'est pas un 45 tours !


maintenant vous le faites.... parler !?!!
Je suis le synthétiseur vocal...
AH...
TILT...
Avec un micro ordinateur on peut faire du dessin industriel, contrôler un processus expérimental, ou une machine-outil.
On peut lui faire taper des lettres, gèrer des stocks, des fichiers, lui faire faire des statistiques.
Cette fois-ci je plonge !..
Hep, Tasci !

L'ORDINATEUR A LA MAISON

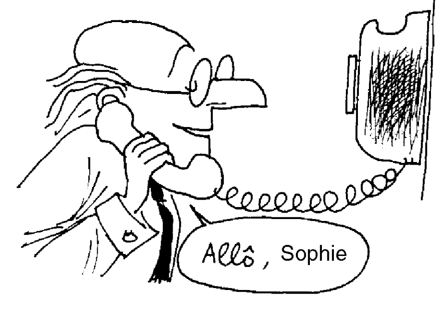


Achille Moneybade soupira profondément.

— Marie Thérèse, c'est affreux. J'ai acheté ce matériel. J'ai passé des heures à tout monter, à tout brancher. Et je ne suis pas plus avancé. Je suis complètement perdu dans ces centaines de pages de notices. Je ne sais même pas par où commencer.

— Pourquoi n'appellez vous pas votre nièce, Sophie. C'est la grosse tête de la famille.

— Vous êtes irremplaçable ! Pourquoi n'y avais-je pas pensé plus tôt !



MÉMOIRES

Oncle Achille, un ordinateur, au départ, c'est comme un énorme paquet de cartes, les CASES MÉMOIRE.

— Et que fait-on avec tout cela ?

— D'abord, nous allons convenir d'un code. Quand je te dirai : « écris telle chose », tu prendra la première carte vierge qui se présentera et tu écriras dessus.

– Mais elles sont toutes vierges, s'écria Moneyback.
– Attends, on a pas encore commencé...
Et si je te dis "efface-tout !" alors tu gommeras ce que tu as mis.
– Bon ....
– Il nous faut maintenant un système de repérage pour ces cartes.
On appellera cela une ADRESSE. D'abord nous utiliserons les lettres de l'alphabet A, B, C, D, ... Nous pourrons aussi utiliser des groupes de deux lettres, ce qui nous fera 26 x 26 = 676 combinaisons possibles.
Ou encore une lettre suivie d'un chiffre, comme A3 ou Z6.
Tout ceci fait partie d'un langage nommé BASIC et dont il sera question ici.

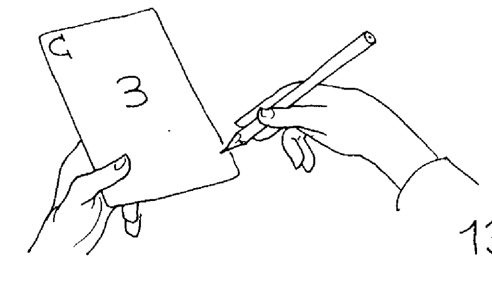
BASIC?..

Tu vois, première idée, on va utiliser un système de repérage littéral des cartes, qui deviendront alors des REGISTRES. le signe = permettra d'indiquer ce qui sera placé dans ce registre. Tu es prêt ?

– Bon, dit Moneyback, je suis prêt, allons-y.
– Attention ! C = 3
Achille prit la première carte qui se présentait, inscrivit un C en haut et à gauche et un 3 au milieu. Puis il replaça la case dans le lot.
Sophie continua en annonçant :

B = 17

A = -8



— Nous avons réalisé une OPÉRATION D'ÉCRITURE DANS LA MÉMOIRE.
— Et comment s'effectue la LECTURE ?
— Je peux te demander "qu'y-a-t-il dans la case, la carte B ?"
— Je chercherai alors la carte qui est marquée d'un B et j'annoncerai son contenu.
— En BASIC cette lecture sera provoquée par l'ordre PRINT ou par l'ordre ? C'est à dire que je dirai PRINT B ou ?B indéfféremment.
Achille se mit au clavier de l'ordinateur.
— Bon, alors pour écrire dans la mémoire je fais B = -17
— Ce n'est pas suffisant, il te faut spécifier la FIN DE MESSAGE. Il y a une touche pour cela. Suivant les ordinateurs elle portera l'inscription ENTER ou Return, que nous appelerons [Fin de Message].
— Et, pour interroger la mémoire, je ferai PRINT B, puis [Fin de Message].
— Exactement...
— Et que se passera-t-il si je fais PRINT Z [Fin de Message] alors que je n'ai pas créé de registre Z ?
— Dans ce cas la l'ordinateur affichera un zéro.
— Il est bizarre, ce zéro !
— En général le zéro des ordinateurs est barré Ø pour le différencier de la lettre o.
Tu peux créer ainsi autant de registres que tu voudras, et décider de les annuler a l'aide l'ordre, de l'INSTRUCTION CLEAR
— Qu'appeles tu une INSTRUCTION.
— Une instruction, c'est une suite de caractères, de signes, qui ont un sens pour l'ordinateur.
— Un mot, quoi ?
— Si tu veux. Ces "mots" du LANGAGE de l'ordinateur s'articulent selon une SYNTAXE rigide.

MESSAGES D'ERREUR

Moneyback, pour s'entrainer, réalisa des opérations variées d'écriture et de lecture. Soudain la machine afficha un MESSAGE D'ERREUR.
(Très souvent ce message est SYNTAX ERROR).

– Qu'est-ce que j'ai fait ? Fouchtre !...

– Tu as fait une ERREUR DE SYNTAXE. Tu as tapé ZM = 5,16 et il n'a pas aimé cela.

– Et alors ?

– Dans le langage des ordinateurs la virgule a une toute autre signification. Ici nous allons devoir suivre des règles anglosaxonnes, et la virgule des nombres décimaux devra être remplacée par un point.

– J'écrirai donc ZM = 5.16

Tout rentra dans l'ordre.

Mais quelques minutes plus tard, nouveau message d'erreur.

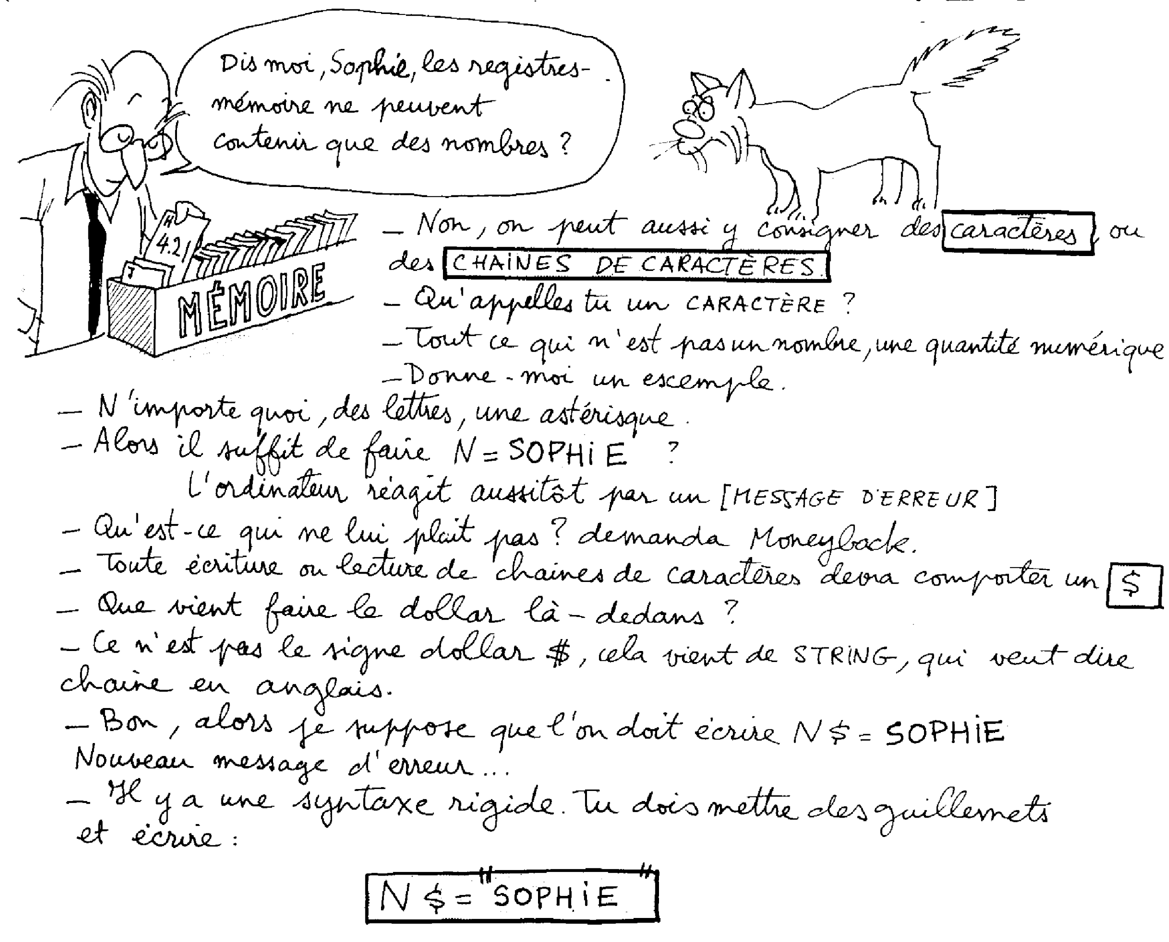
– Ici tu as tapé 30 au lieu de 3∅, c'est classique.

– Ah oui, ce zéro bizarre des informaticiens ! (*)

– Il faut bien un moyen quelconque pour différencier la lettre O du caractère représentant le zéro.

(*) IL FAUT BIEN DIFFÉRENCIER LA LETTRE O ET LE ZÉRO D'UNE MANIÈRE QUELCONQUE.
DANS LES MACHINES JAPONAISE, C'EST L'INVERSE. LE ZÉRO S'ÉCRIT O ET LA LETTRE O S'ÉCRIT ∅

CHAINES DE CARACTÈRES



— Non, on peut aussi y consigner des caractères, ou des CHAINES DE CARACTÈRES

— Qu'appelles tu un CARACTÈRE ?

— Tout ce qui n'est pas un nombre, une quantité numérique

— Donne-moi un escemple.

— N'importe quoi, des lettres, une astérisque.

— Alors il suffit de faire N = SOPHIE ?

L'ordinateur réagit aussitôt par un [MESSAGE D'ERREUR]

— Qu'est-ce qui ne lui plait pas ? demanda Moneyback.

— Toute écriture ou lecture de chaines de caractères devra comporter un $

— Que vient faire le dollar là-dedans ?

— Ce n'est pas le signe dollar $, cela vient de STRING, qui veut dire chaine en anglais.

— Bon, alors je suppose que l'on doit écrire N$ = SOPHIE

Nouveau message d'erreur...

— Il y a une syntaxe rigide. Tu dois mettre des guillemets et écrire :

```
N$ = "SOPHIE"
```

La lecture se faisant avec `PRINT N$`

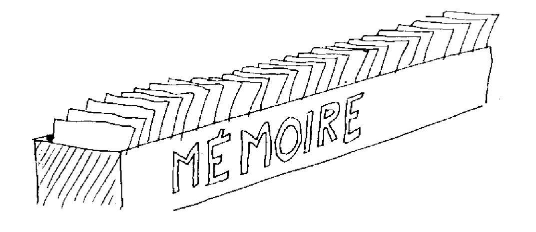
ROM ET RAM

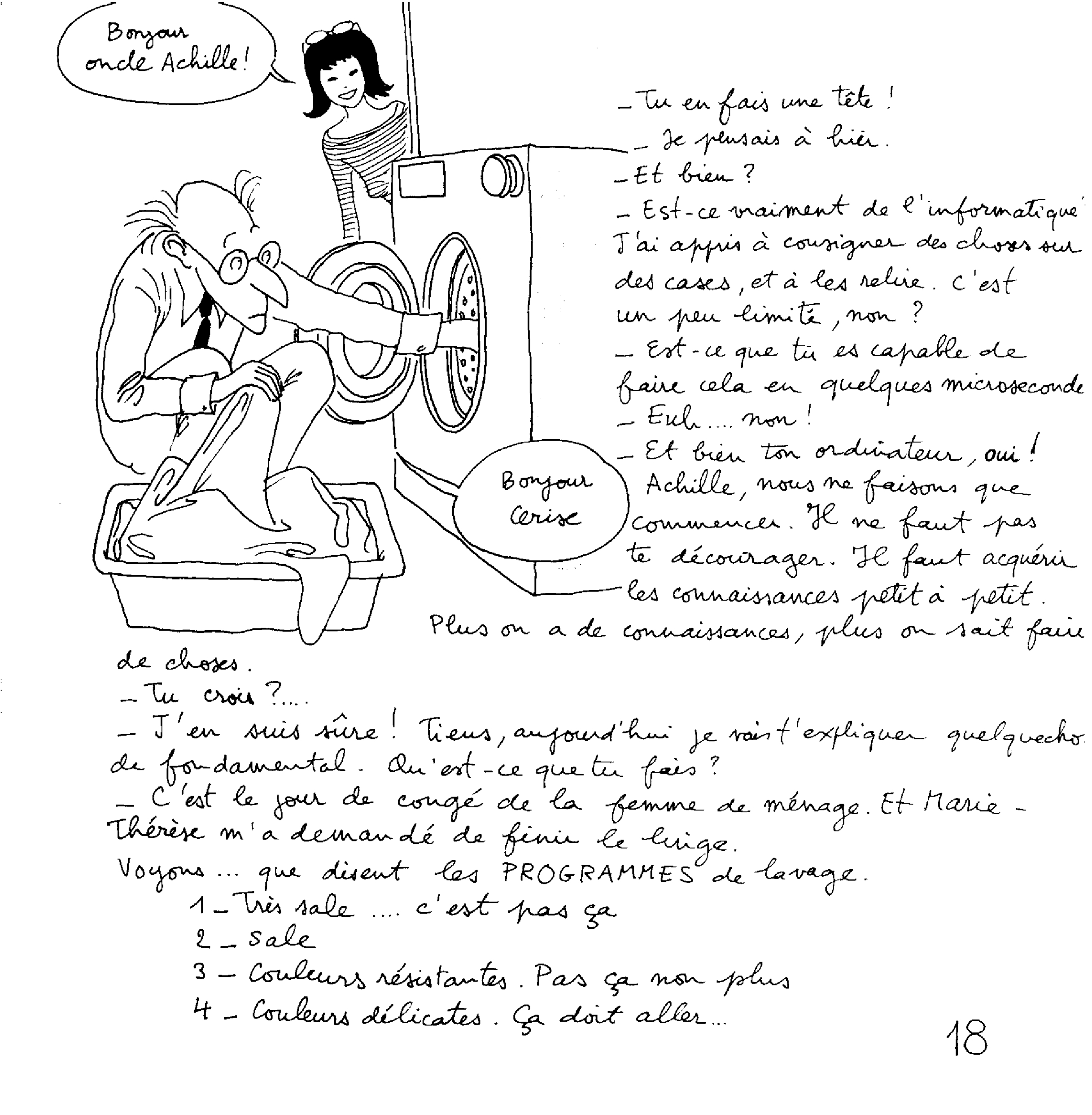
Sophie revint au paquet de cartes.
– L'ordinateur peut contenir beaucoup de choses variées. Certaines de ces choses sont effaçables et d'autres non. C'est à dire qu'il n'existe aucun ordre, aucune instruction, qui permette de modifier ces contenus.
Ces éléments de mémoire où on ne peut écrire et qu'on ne peut effacer s'appellent des ROM (read only memory). Ce qui signifie en anglais "mémoire qu'on ne peut que lire".
Par opposition les mémoires où l'utilisateur pourra écrire à sa guise s'appeleront RAM (random access memory) c'est à dire mémoire où l'on peut accéder à n'importe quel moment.

Voilà maintenant c'est ineffaçable...

– Bon, dit Moneyback, je ne sais pas trop où tout cela me mènera, mais j'en ai assez appris pour aujourd'hui.
Sophie promit de revenir le lendemain. Achille resta longtemps songeur devant le classeur contenant le paquet de cartes.







- Tu en fais une tête !
- Je pensais à hier.
- Et bien ?
- Est-ce vraiment de l'informatique ? J'ai appris à consigner des choses sur des cases, et à les relier. C'est un peu limité, non ?
- Est-ce que tu es capable de faire cela en quelques microsecondes ?
- Euh.... non !
- Et bien ton ordinateur, oui ! Achille, nous ne faisons que commencer. Il ne faut pas te décourager. Il faut acquérir les connaissances petit à petit. Plus on a de connaissances, plus on sait faire de choses.
- Tu crois ?....
- J'en suis sûre ! Tiens, aujourd'hui je vais t'expliquer quelquechose de fondamental. Qu'est-ce que tu fais ?
- C'est le jour de congé de la femme de ménage. Et Marie-Thérèse m'a demandé de finir le linge.

Voyons ... que disent les PROGRAMMES de lavage.

1 - Très sale .... c'est pas ça
2 - sale
3 - Couleurs résistantes. Pas ça non plus
4 - Couleurs délicates. Ça doit aller...

LA NOTION DE PROGRAMME

– Bon. J'ai mis la lessive dans les casiers ad hoc.
C'est une machine perfectionnée. Quand j'ai choisi le type de programme un voyant s'allumme et m'indique où il faut mettre la lessive. Puis je vois apparaître le message "mettez le linge dans le tambour". Si je n'ai pas bien refermé la porte, ma machine émet un son et me le rappelle. Quand tout est OK, elle démarre. Elle a neuf programmes de lavage.
– C'est comme l'ordinateur.
– Qu'est-ce que tu veux dire ?!?
– Lui aussi a des programmes.
– Ça par exemple ! tu veux dire qu'on peut lui faire laver le linge !!!
– Euh .... oui et non .... on peut lui faire faire beaucoup de choses.
Cerise prit un paquet de cartes dans le classeur.
– Voyons... quelles sont les opérations liées au programme de lavage. Cela doit être sur la notice. Ah ... voila ! programme couleurs délicates : Remplissage du tambour avec de l'eau, chauffage à 40°, prélèvement de la lessive disposée dans les casiers, lavage, cinq rinçages...



Il y a six programmes de lavage.

Sophie prit le paquet de cartes. Et elle écrivit :

4Ø Lavage couleurs délicates

41 Remplissage du tambour avec de l'eau

42 chauffage à 40°

43 Prélèvement lessive

44 Lavage

45 cinq rinçages

46 FIN

Et maintenant je les mets dans la mémoire

– Qu'est-ce que c'est que ces numéros, demanda Achille.

– Ils ordonnent la séquences d'opérations, d'instructions, qui constituent ce **PROGRAMME**.

– Un programme est une **SUITE ORDONNÉE D'INSTRUCTIONS ?**

– Avec d'autres choses que nous verrons plus loin. Mais au départ c'est cela.

– Mais pourquoi as-tu commencé la numérotation à 4Ø ?

– J'aurais pu commencer par n'importe quel numéro. Et j'aurais pu numéroter les cartes de deux en deux, de dix en dix. Cela n'a aucune importance, seul l'ordre compte.

J'ai commencé par le chiffre 4Ø parce qu'il s'agissait du programme quatre.

– Je devine qu'entre 1Ø et 2Ø tu pourrais loger le programme numéro un, entre 2Ø et 3Ø le programme numéro trois, etc...

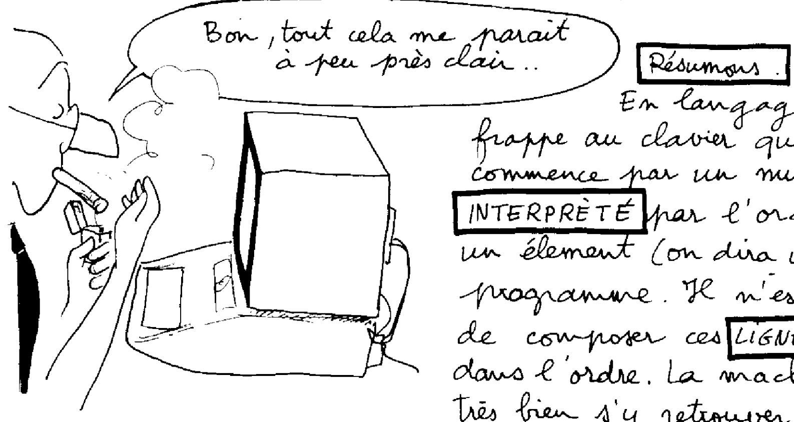
– Bien sûr, je peux mettre plusieurs programmes dans ma mémoire. Et je peux les faire exécuter à la demande.

– Comme moi avec ma machine à laver.

– comme tu as vu, à la fin, j'ai mis une [INSTRUCTION D'ARRÊT.] Dans un ordinateur, si cette instruction n'est pas présente, la machine exécutera toute instruction repérée avec un numéro plus grand.

– Je suppose que dans la machine il doit y avoir quelquechose d'analogue.

– sinon, en lançant par exemple le programme deux, l'ordinateur, après exécution, enchaînerait par le trois, le quatre, etc..



Résumons.

En langage BASIC, si je frappe au clavier quelquechose qui commence par un numéro, ceci sera INTERPRÈTÉ par l'ordinateur comme un élément (on dira une ligne) d'un programme. Il n'est pas nécessaire de composer ces LIGNES-PROGRAMME dans l'ordre. La machine saura très bien s'y retrouver





LE PREMIER PROGRAMME DE MONEYBACK

– Sophie, combien y a-t-il d'instructions dans un programme ?
– Autant que tu voudras, compte tenu de la capacité de la mémoire.
– Ce que je voulais dire, c'est : quel en est le nombre minimum ?
– Une. Tiens, tu peux par exemple imaginer un programme ne comportant qu'une seule instruction END [instruction d'arrêt]. et qui s'écrirait :

1 END

Pour EXECUTER UN PROGRAMME, LANCER ou APPELER un PROGRAMME (expressions équivalentes), on doit composer RUN (et, bien sûr, appuyer sur la touche [Fin de message].
– Et que se passe-t-il ?
– L'ordinateur exécute e seul ordre présent qui est... de s'arrêter !
– Hum... il y a des gens qui, dans la vie, agissent ainsi.
– On peut continuer à s'amuser. Tiens, regarde cet autre programme.

1 RUN

– L'instruction RUN, dans certains ordinateurs, est PROGRAMMABLE. C'est à dire qu'on peut l'inclure dans un programme.
– Là, le gars se dit : « allons-y .... allons-y .... » à l'infini. Et en fait il ne fait rien !



– Soyons sérieux. Tu peux inclure dans un programme des ordres d'impression PRINT

– Comment procède-t-on ?

– La règle de syntaxe est la suivante : tu inscris un PRINT dans ta LIGNE PROGRAMME. Puis tu ouvres les guillemets. Tu inscris ce que tu veux (sauf des guillemets), et tu n'oublies pas de refermer les guillemets.

– Sinon .... SYNTAX ERROR !

Achille composa :

```
1 PRINT "CLAVIER EST UN ANE"
```

Puis il lança ce programme à l'aide de RUN. La phrase s'inscrivit sur l'écran.

– Et, pour effacer ?

– Il existe une instruction [Effaçage Ecran], qui diffère d'un ordinateur à l'autre. On peut l'inclure dans un programme (ou l'utiliser directement)

```
1 [Effaçage Ecran]
2 PRINT "CLAVIER EST UN ANE"
```

– Comme cela il y a d'abord nettoyage de l'écran, puis impression de la phrase. Sophie, combien y a-t-il d'instructions ?

– L'ensemble des instructions constitue le LANGAGE de l'ordinateur. Il y en a évidemment pas mal. Apprendre à programmer, c'est se familiariser avec ce jeu d'instructions, à les combiner (SYNTAXE). La qualité d'un langage réside en particulier dans la variété et la souplesse de son jeu d'instructions.

Au cours de notre apprentissage, nous introduirons les **INSTRUCTIONS** au fur et à mesure des besoins. L'instruction **NEW** a pour effet de détruire, d'effacer tout programme dans la mémoire.

– Est-elle programmable ?

– Oui, ainsi tu peux créer des programmes qui, après utilisation, se détruisent eux-mêmes. Il suffit de créer en fin de programme une LIGNE portant l'ordre NEW (New veut dire nouveau, neuf)

```
1 [Effaçage Ecran]
2 PRINT "ATTENTION, CE PROGRAMME VA SE
DETRUIRE LUI-MEME !"
3 NEW
```



Moneyback composa NEW d'abord, au clavier, pour effacer toute trace du programme précédent (ce qu'il est conseillé de faire systématiquement, surtout sur les machines de poche qui peuvent conserver un programme en mémoire même quand elles sont éteintes)
Puis il écrivit d'une traite le programme ci-après :

```
1 [Effaçage Ecran]
2 PRINT"MONSIEUR,"
3 PRINT"J'AI BIEN RECU VOTRE HONOREE DU 11 COURANT"
4 PRINT"ET JE VOUS EN REMERCIE"
5 PRINT"QUANT A NOTRE ARTICLE CP 9013-A, NOUS NE"
6 PRINT"L'AVONS PLUS EN STOCK,HELAS."
7 PRINT"NOUS DEVRIONS POUVOIR PALLIER CETTE"
8 PRINT"RUPTURE DE STOCK EN DATE DU 2"
9 PRINT"MARS 1982"
10 PRINT"AVEC NOS EXCUSES "
11 PRINT"          ACHILLE  MONEYBACK"
12 END
```

Moneyback fit RUN, et vit apparaître :

MONSIEUR,
J'AI BIEN RECU VOTRE HONOREE DU 11 COU
RANT.
ET JE VOUS EN REMERCIE.
QUANT A NOTRE ARTICLE CP9013-A,NOUS NE
L'AVONS PLUS EN STOCK,HELAS.
NOUS DEVRIONS PALLIER
CETTE RUPTURE DU STOCK EN DATE DU 2
MARS 1982.
AVEC NOS EXCUSES.
ACHILLE MONEYBACK

C'EST PAS MAL,MAIS À LA LIGNE2, TU AS VOULU TROP EN METTRE

Sophie se mit au clavier :

– Attends, on va arranger cela. Elle fit NEW, et composa à son tour :

```
10 [Effaçage Ecran]
20 VTAB5 : HTAB16 : PRINT"MONSIEUR,"
30 PRINT:PRINT"J'AI BIEN RECU VOTRE HONOREE DU ONZE"
40 PRINT:PRINT"COURANT,ET JE VOUS EN REMERCIE" : PRINT
50 PRINT"QUANT A NOTRE ARTICLE 9013-A, NOUS NE "
60 PRINT: PRINT"L'AVONS PLUS EN STOCK,HELAS."
70 PRINT: PRINT" NOUS DEVRIONS PALLIER CETTE"
80 PRINT: PRINT" RUPTURE DE STOCK EN DATE DU DEUX"
90 PRINT: PRINT"MARS 1982" : PRINT
100 HTAB8: PRINT" AVEC NOS EXCUSES"
110 PRINT: HTAB17: PRINT "ACHILLE MONEYBACK"
120 END
```

MONSIEUR,

J'AI BIEN REÇU VOTRE HONORÉE DU ONZE COURANT, ET JE VOUS EN REMERCIE.

QUANT À NOTRE ARTICLE CP9013-A, NOUS NE L'AVONS PLUS EN STOCK, HELAS.

NOUS DEVRIONS POUVOIR PALLIER CETTE RUPTURE DE STOCK EN DATE DU DEUX MARS 1982.

AVEC NOS EXCUSES,

ACHILLE MONEYBACK

Formidable ! Comment as-tu fait cela ?

A la ligne 20 j'ai mis des INSTRUCTIONS DE TABULATION. Elles peuvent différer d'un ordinateur à l'autre, mais en voici le principe : Mon écran d'affichage comprend un certain nombre de lignes. Et chaque ligne est susceptible de recevoir un certain nombre de CARACTÈRES (y compris les blancs).

VTAB5 signifie qu'on n'écrira qu'à partir de la 5e ligne. Et HTAB16 veut dire : n'écrire qu'à partir de la seizième case de la ligne.

– Et les deux points : ?

– Ah, c'est un truc qui permet de loger deux instructions sur la même ligne.

– Et le PRINT en début de ligne 3Ø ?
– Il donne une ligne vide, un blanc, à l'affichage. C'est pour la présentation. Sinon le texte est un peu serré.
– Est-ce que j'ai un moyen de repérer un programme, de lui donner un nom, par exemple, qui me permette de m'y retrouver ?

– Tu peux consigner des REMARQUES dans un programme BASIC, à l'aide de l'instruction **REM**
– Mais comment allons nous MODIFIER ce programme maintenant qu'il est composé ? Il va falloir tout reprendre à zéro ?
– Pas du tout ! Comme tu vois j'ai numéroté les lignes de dix en dix ce qui me laisse de la place pour en rajouter. Il me suffit de taper :

```
5REM  LETTRE SUR RUPTURE STOCK
7REM  ARTICLE  CP 9013-A
```

– Mais, comment peux-tu vérifier que ces deux lignes ont bien été incluses dans le programme ?
– A tout moment, je peux provoquer l'affichage, sur l'écran, du programme en utilisant l'ordre, la commande **LIST**
Mais je peux "LISTER" une ligne particulière en faisant **LIST 3Ø**
Ou même une suite de lignes en composant : **LIST 5, 8Ø**
(lignes 5 à 8Ø, incluses)

comment utiliser l'imprimante ?

Il y a des commandes spéciales. Cela dépend des modèles. Une fois l'imprimante "activée", ce qui apparait sur l'écran est également imprimé sur papier.

– Si je fais une erreur en composant une ligne, qu'est ce que je dois faire ?

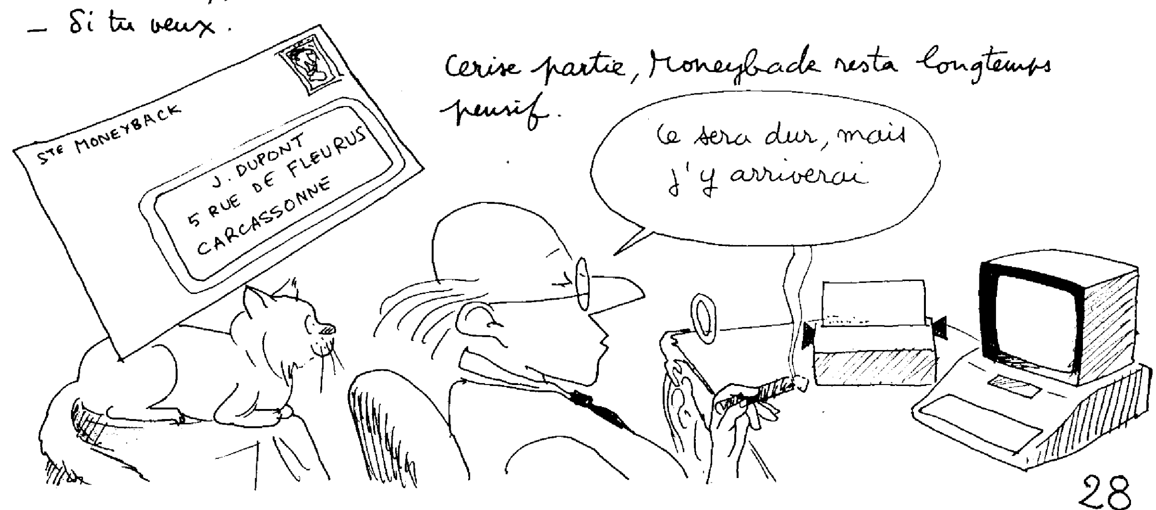
– Sur le clavier, tu as d'abord deux touches [←] et [→]. La première permet un retour en arrière. Ce qui te permet de corriger caractère par caractère. En utilisant la barre d'espacement, tu imprimes un BLANC, un caractère "VIDE", ce qui permet l'effacement.

Pour annuler une ligne, le plus simple est de composer son numéro puis de presser la touche [FIN DE MESSAGE]. Pour remplacer une ligne par une autre, cette opération préalable n'est pas nécessaire. Il suffit de réécrire cette ligne, puis de presser la touche [FIN DE MESSAGE].

Pour effacer une suite de lignes, tu utilises [DEL 5Ø, 11Ø] (ceci signifie : annuler les lignes 5Ø à 11Ø, incluses).

– Dis moi, en utilisant astucieusement les instructions HTAB et VTAB je peux placer sur une feuille l'adresse de mon correspondant, au bon endroit, pour qu'elle puisse se lire à travers la fenêtre transparente d'une enveloppe.

– Si tu veux.



Moneyback se mit au clavier, se trompa de nombreuses fois, corrigea ses erreurs. Quand Marie-Thérèse rentra, il lui dit :

– Ma douce, viens voir.

– Voir cette affreuse machine !

– Viens, je t'en prie, pour me faire plaisir.

– Bon...

– Maintenant, composes RUN et presse sur cette touche [FIN DE MESSAGE]

La machine afficha aussitôt :

MA BICHETTE,

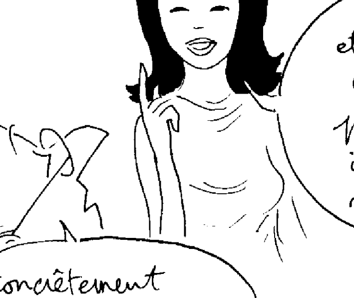
VEUX TU QUE JE T'EMMENE AU RESTAURANT CE SOIR ?

TON GROS LAPIN.



PROGRAMMES AVEC ENTRÉES DE DONNÉES

Sophie trouva son oncle très perplexe.
– J'ai eu une fausse joie hier. Je croyais que tout cela allait pouvoir me servir à quelquachose. Mais, regarde ce programme qui écrit des lettres. Je ne peux même pas modifier le numéro de l'article.



Il faut, continua Sophie, spécifier quel type de REGISTRE l'ordinateur devra créer pour mettre cette donnée.

`INPUT N` mettra l'ordinateur en attente d'une donnée numérique.
`INPUT N$` mettra l'ordinateur en attente d'une chaine de caractères.
Et tu peux aussi faire poser, en clair, la question par la machine.
Exemple :

```
2ø INPUT"DATE LETTRE CLIENT :"; D$
```

Attention ⇒ n'oublier ni les guillemets, ni le point virgule.

(*) INPUT : en anglais, signifie ENTRÉE

- Et on peut tout mettre dans les guillemets ?

- Sauf des guillemets ....

Moneyback modifia le programme précédent, comme suit :

```
5 REM LETTRE SUR RUPTURE DE STOCKS
1Ø [Effaçage Ecran]
2ØINPUT"DATE LETTRE CLIENT:";D$
3ØINPUT"REFERENCE NOTRE ARTICLE:";AR$
4ØINPUT"DATE NOTREREPONSE:";R$
5ØPRINT"DATE A LAQUELLE LE CLIENT POURRA ETRE"
55 PRINT
6Ø INPUT"LIVRE:";LI$
7Ø HOME
8Ø[Activation de l'Imprimante]
9Ø PRINT"SOCIETE MONEYBACK"
1ØØHTAB 2Ø :PRINT"LE ";R$
11ØVTAB5:HTAB16:PRINT"MONSIEUR,"
12ØPRINT:PRINT"BIEN RECU VOTRE LETTRE DU ";D$:PRINT
13ØPRINT"DONT NOUS VOUS REMERCIONS.":PRINT
14ØPRINT"QUANT A L'ARTICLE ";AR$;" ,NOUS NE ":PRINT
15ØPRINT"L'AVONS PLUS EN STOCK,HELAS.": PRINT
16ØHTAB11:PRINT"NOUS POURRIONS PALLIER CETTE":PRINT
17ØPRINT"RUPTURE DE STOCK VERS LE ";LI$ :PRINT
18ØHTAB8:PRINT"AVEC NOS EXCUSES,":PRINT
19ØHTAB17 :PRINT"ACHILLE MONEYBACK"
2ØØ[DESACTIVATION DE L'IMPRIMANTE]
21Ø END
```

Plusieurs remarques. A la ligne 14Ø le point virgule permet d'enchainer des impressions (sur l'écran et sur le papier), que les choses à imprimer soit inscrites directement dans le programme ou présentes dans des registres (ici AR$). Le PRINT de séparation de ligne peut être mis indifférement en début ou en fin de ligne.

Si on ne dispose pas des instructions VTAB et HTAB on peut remplacer la première par une série de PRINT : PRINT : PRINT (chaque PRINT entraîne un saut de ligne), et la seconde par des "

Moneyback fit RUN et sur l'écran apparut :

DATE LETTRE CLIENT ▨

curseur clignotant

– Bon, qu'est-ce que je fais ?

– Bien, tu tapes la date, par exemple 12 SEPTEMBRE. Puis tu appuies sur la touche [FIN DE MESSAGE].

Moneyback répondit ainsi aux différentes questions, et la machine imprima :



A l'aide de ce premier programme Achille venait d'entrer de plein pied dans la jungle de l'informatique.

STOCKER, CHARGER UN PROGRAMME

– Mon ordinateur est donc prêt à m'écrire autant de lettres que je le veux.
– Tu mettras le premier type de lettre de la ligne 5 à la ligne 210 puis la seconde de la ligne 1000 à la ligne 1320 par exemple. La troisième de la ligne 2000 à .... etc. Et tu "appeleras" ces différents programmes en faisant RUN5 ou RUN1000 ou RUN 2000, etc...
– Tous les problèmes sont résolus...
– Tu crois ? Ton ordinateur est immobilisé. Il ne peut plus rien faire d'autre. Et si quelqu'un, en ton absence, éteint la machine, ou que survienne une coupure de courant : H, envolé, ton programme (à moins que ton ordinateur n'ait une MÉMOIRE NON VOLATILE.)-
Il va te falloir stocker, RECOPIER, ce programme dans une MÉMOIRE EXTERNE..
Il y a donc plusieurs types de mémoires ?



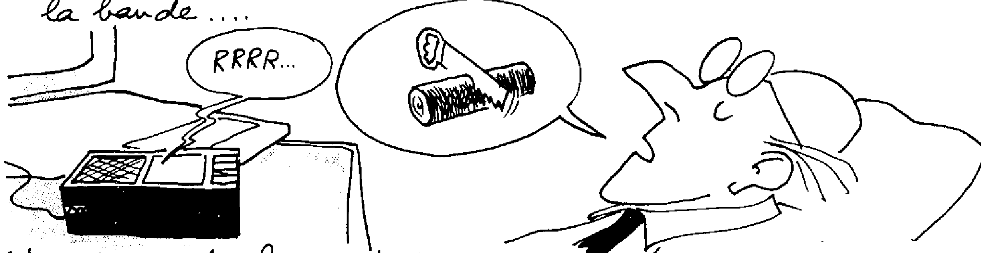
– Et comment ! Notre micro ordinateur a tout d'abord sa MÉMOIRE CENTRALE avec ses ROM (son LOGICIEL DE BASE). et ses RAM, où s'inscrivent les programmes et les données.
Il peut être relié à deux types de mémoires externes, une mémoire à cassette et une mémoire à DISKETTES, plus commode.

– Commençons par le magnétophone. Le système manque de souplesse.
Il faut d'abord positionner la bande, en s'aidant du compteur.
Ne serait-ce que pour trouver une place vide sur celle-ci.
Un ordre clavier (CSAVE) provoque la recopie du programme, codé en binaire, qui se trouve transféré sur la bande sous forme d'une suite d'impulsions.

La recherche du programme est laborieuse. On se positionne à l'aide du compteur du magnétophone. Sur certains ordinateurs l'ordre de stockage est simplement CSAVE et l'ordre de chargement CLOAD (voir notice).
Le CHARGEMENT est une opération où ce qui est présent sur la bande va se trouver recopié en memoire centrale. Cette recopie n'altère en rien le contenu de cette mémoire périphérique. Même remarque pour l'opération de STOCKAGE (qui n'altère pas ce qui est en mémoire centrale).
Certains systèmes permettent de repérer le programme à l'aide d'un IDENTIFICATEUR de quelques lettres

Exemple : CSAVE "LETTRES" (*)
CLOAD "LETTRES"

Dans ces conditions, en explorant la bande, l'ordinateur est capable de trouver lui-même le programme recherché.
Mais on reste tributaire de la vitesse, assez lente, de défilement de la bande ....



(*) SAVE, en Anglais, veut dire SAUVER, préserver.
LOAD veut dire CHARGER

le système à disques souples, DISKETTES, est infiniment plus commode, rapide et souple.

les ordres "diskettes" sont :

SAVE PROGRAMME LETTRES

LOAD PROGRAMME LETTRES

Sans guillemets.

Lorsqu'on compose CATALOG, la machine donne immédiatement les programmes contenus sur cette face de la diskette.



misère...

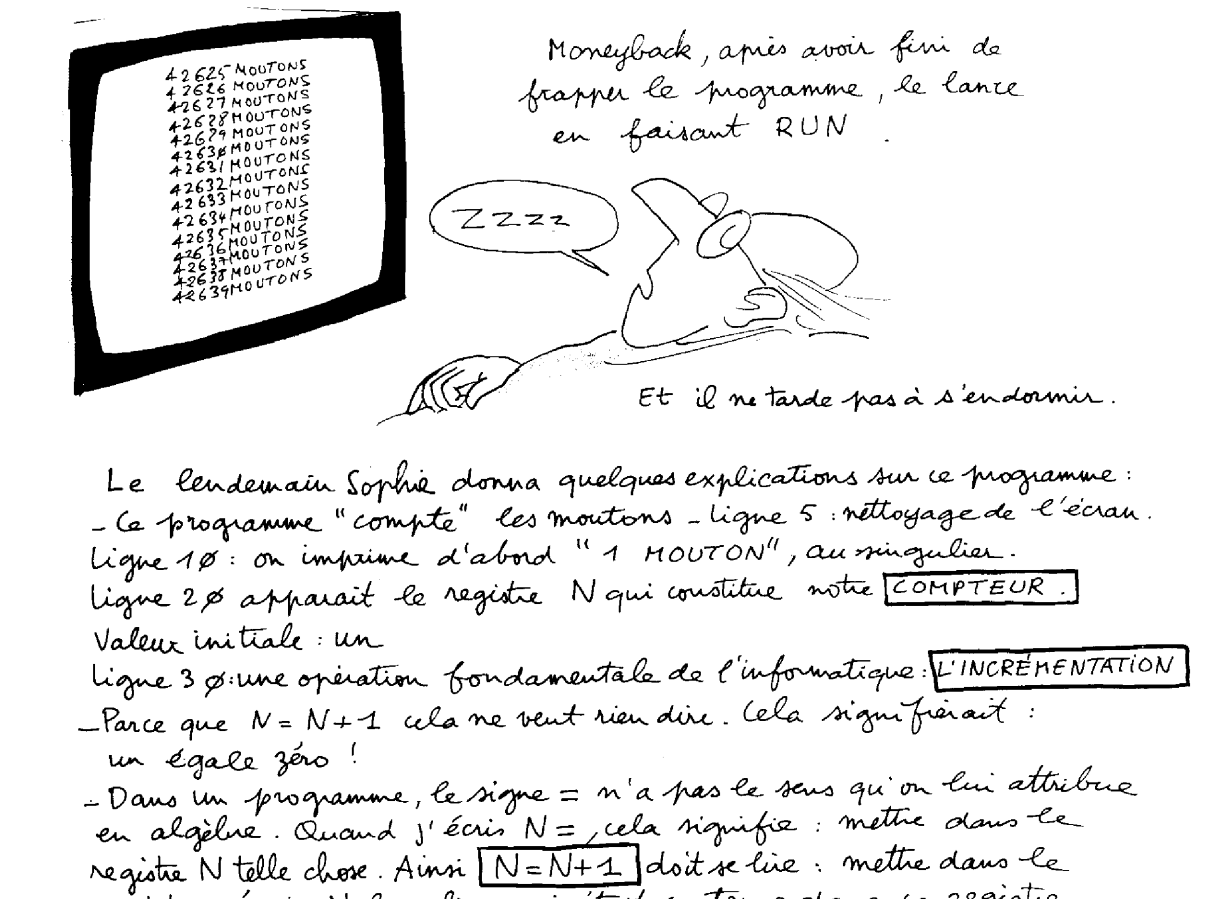


BOUCLES






Pfff .... une heure et quart.
c'est l'insomnie !
Allô, Sophie, excuse-moi de t'appeler si tard. Je n'arrive pas à dormir. C'est peut être la conjoncture ?
Ecoute, oncle Achille, je vais te dicter un programme qui devrait t'aider. Tu n'as qu'à noter ...
Alors, tu dis :
5 [Effaçage Ecran]
1Ø PRINT "1 MOUTON"
2Ø N = 1
3Ø N = N + 1
4Ø PRINT N; "MOUTONS"
5Ø GOTO 3Ø
N = N + 1
cela semble absurde !..
enfin ....



Le lendemain Sophie donna quelques explications sur ce programme :

– Ce programme "compte" les moutons – ligne 5 : nettoyage de l'écran.
Ligne 1Ø : on imprime d'abord "1 MOUTON", au singulier.
Ligne 2Ø apparaît le registre N qui constitue notre **COMPTEUR.**
Valeur initiale : un
Ligne 3Ø : une opération fondamentale de l'informatique : **L'INCRÉMENTATION**

– Parce que N = N+1 cela ne veut rien dire. Cela signifierait : un égale zéro !

– Dans un programme, le signe = n'a pas le sens qu'on lui attribue en algèbre. Quand j'écris N =, cela signifie : mettre dans le registre N telle chose. Ainsi **N=N+1** doit se lire : mettre dans le registre mémoire N la valeur qui était contenue dans ce registre accrue d'une unité. On devrait écrire N ⇐ N+1.

– Alors, incrémenter, c'est quand on accroît un registre d'une unité ?

– Non, tu peux incrémenter d'une quantité quelconque, entière ou décimale, qu'on appellera **L'INCRÉMENT.** Tu peux aussi **DÉCRÉMENTER.**
Par exemple **I = I – 3.6Ø1**

– A la ligne 4Ø tu retrouves un ordre d'impression qui affiche bout à bout N , c'est à dire le nombre de moutons, suivi de "MOUTONS". Et en 5Ø tu as une instruction de BRANCHEMENT INCONDITIONNEL, qui te renvoit à la ligne 3Ø. GOTO, en anglais, veut dire "aller vers". Ce que faisant, on a créé une BOUCLE SANS FIN.

– Effectivement, on est déja à plus de cent mille moutons. Comment fait-on pour stopper un tel programme?

– Cela dépend des machines. Il existe une [INSTRUCTION D'ARRET] que l'on compose au clavier. L'ordinateur alors s'arrête net, et affiche la dernière ligne exécutée.



– Dis moi, Sophie, tout à l'heure, ça allait beaucoup trop vite. Au moins une ligne imprimée sur l'écran à chaque seconde. On pouvait à peine lire. Peut-on ralentir cela ?

– Oui, mais je te montrerai cela plus tard.

– Mais je vais te montrer comment on peut compter un nombre donné de moutons. Pour cela il faut utiliser une INSTRUCTION DE BRANCHEMENT CONDITIONNEL

– Montre moi cela !

BRANCHEMENTS CONDITIONNELS

– On utilise l'instruction IF... THEN... (SI.... ALORS...)
Supposons qu'on veuille compter cent cinquante moutons. On écrira :


```
5 [Effaçage Ecran]
10 PRINT "1 MOUTON"   (*)
20 N=1
30 N=N+1
40 PRINT N ; " MOUTONS"
50 IF N=150 THEN 70
60 GOTO 30
70 END
```

A chaque fois que l'on repassera par la ligne 50 un TEST sera effectué. Tant que N est différent de 150 le bouclage continuera. Mais si N = 150 alors il est prescrit d'aller en ligne 70, et on réalise une SORTIE DE BOUCLE. En 70 on a mis une instruction d'arrêt END. Alors le travail s'arrête.

– Ah, dit Achille, j'ai une autre idée. On peut utiliser une instruction INPUT pour préciser, avant exécution, combien de cycles sont à effectuer

– Vas-y

(*) Sur les machines de poche une instruction PAUSE affiche pendant environ une seconde les caractères entre guillemets

Moneyback composa le programme suivant :

```
5 [Effaçage Ecran]
7 INPUT "COMBIEN DE MOUTONS ? "; NM
9 [Effaçage Ecran]
10 PRINT " 1  MOUTON"
20 N = 1
30 N = N + 1
40 PRINT  N; "  MOUTONS"
50 IF N = NM  THEN 70
60 GOTO 30
70 END
```

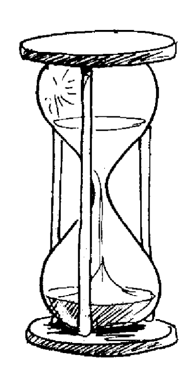
— Bien ! mais il y a une façon plus compacte d'écrire cela :

BOUCLES FOR... NEXT

```
5 [Effaçage Ecran]
7 INPUT "COMBIEN DE MOUTONS ? "; NM
9 [Effaçage Ecran]
10 PRINT "1  MOUTON"
20 FOR N = 2 TO NM
30 PRINT  N; "  MOUTONS"
40 NEXT N
50 END
```

— Tu vois, ça se fait tout seul. La ligne 20 signifie " pour N allant de 2 à NM ". La ligne 40 signifie " Valeur suivante de N ". Et maintenant je vais pouvoir te montrer comment on peut régler le temps d'affichage non permanent sur l'écran.

BOUCLES DE TEMPORISATION



L'informatique permet de simuler un comportement, très courant, qui consiste à refaire la même chose éternellement. En psychanalyse ceci s'appelle la compulsion de répétition. L'ordinateur peut rééditer éternellement une tâche sans aucun contenu. Par exemple, à travers le programme :

```
1 GOTO 1
```

Pour "temporiser" on introduira à l'endroit voulu le nombre ad hoc de cycles vides. On peut loger la boucle sur une seule ligne. Par exemple, dans le programme précédent, on pourrait caser :

```
35 FOR I=Ø TO 5ØØ : NEXT I
```

Le délai introduit est évidemment proportionnel au nombre de boucles.

Bon sang, avec tout cela, je devrais être capable d'articuler des programmes déja plus élaborés

INCRÉMENTATION

IF... THEN

PRINT

STOCKAGE

END

RUN

NEW

INPUT

TEMPORISATION

Boucles FOR... NEXT

Moneyback se demanda si, en utilisant ce qu'il savait déja, il ne pourrait pas mettre sous forme d'un programme le calcul de l'impôt sur le revenu.

– Voyons, on part du revenu imposable R et du nombre de parts N. Deux informations qui pourront être données à la machine grâce à des instructions INPUT.

On divise R par N et on obtient ainsi le quotient familial QF.
le calcul de l'impôt dépend alors de la valeur de QF. Sur la feuille de l'administiation, Achille lut :

Votre quotient QF :	T taux de l'impôt	Calcul de l'Impôt I
Si QF < ou = à 9890F	0%	I = 0
Si QF>9890F et QF<=10340F	5%	I = (R×0,05) – (494,5 ×N)
Si QF>10340F et QF<=12270F	10%	I = (R×0,10) – (1011,5 ×N)
Si QF>12270F et QF< 19410F	15%	I = (R×0,15) – (1625 ×N)
Si QF>19410F et QF <= 24950F	20%	I = (R×0,20) – (2595,5 ×N)
Si QF>24950F et QF <= 31360F	25%	I = (R×0,25) – (3843 ×N)
Si QF> 31360F et QF <=37940F	30%	I = (R×0,30) – (5411 ×N)
Si QF>37940F et QF<=43770F	35%	I = (R×0,35) – (7308 ×N)
Si QF>43770F et QF<= 72940F	40%	I = (R×0,40) – (9496,5 ×N)
Si QF >72940F et QF <= 100320F	45%	I = (R×0,45) – (13143 ×N)
Si QF>100320F et QF<= 118660F	50%	I = (R×0,5) – (18159,5 ×N)
Si QF >118660F et QF <= 135 000F	55%	I = (R×0,55) – (24092,5 ×N)
Si QF >135 000F	60%	I = (R×0,6) – (30842,5 ×N)

ouf...

CALCULER SES IMPOTS



Bon, se dit Achille, chaque ligne de ce tableau figure une TRANCHE D'IMPOSITION
Le début n'est pas difficile. Je n'ai qu'à mettre :

```
10 [Effaçage Ecran]
20 PRINT "CALCUL D'IMPOT SUR LE REVENU"
30 PRINT
40 INPUT "VOTRE REVENU IMPOSABLE :"; R : PRINT
50 INPUT "NOMBRE DE PARTS :"; N : PRINT
60 QF = R/N : REM QUOTIENT FAMILIAL
70 PRINT "VOTRE QUOTIENT FAMILIAL :"; QF
```

Pour la suite, je devrais pouvoir m'en tirer avec une série de .... IF.... THEN....

Sophie vint à la rescousse :
– Formidable, mon oncle, tu fais des progrès ! Je vais t'aider un peu.
L'instruction IF.... THEN... se base sur un test, mais tu peux donner à ce test plusieurs facettes, en utilisant le "ET" et le "OU", ce qui te donnera SI CECI ..... ET CELA .... ET CELA ... ALORS ....
ou : SI CECI .... OU CELA ..... ALORS ....
"ET" en anglais se dit AND
"OU" se dit "OR"
Dans ces conditions, voilà comment s'écrirait par exemple la deuxième ligne du tableau du calcul de l'impôt :

```
IF QF>989Ø [AND] QF<=1Ø34Ø THEN T=5 : I = R*Ø.Ø5 - 494.5 *N
```

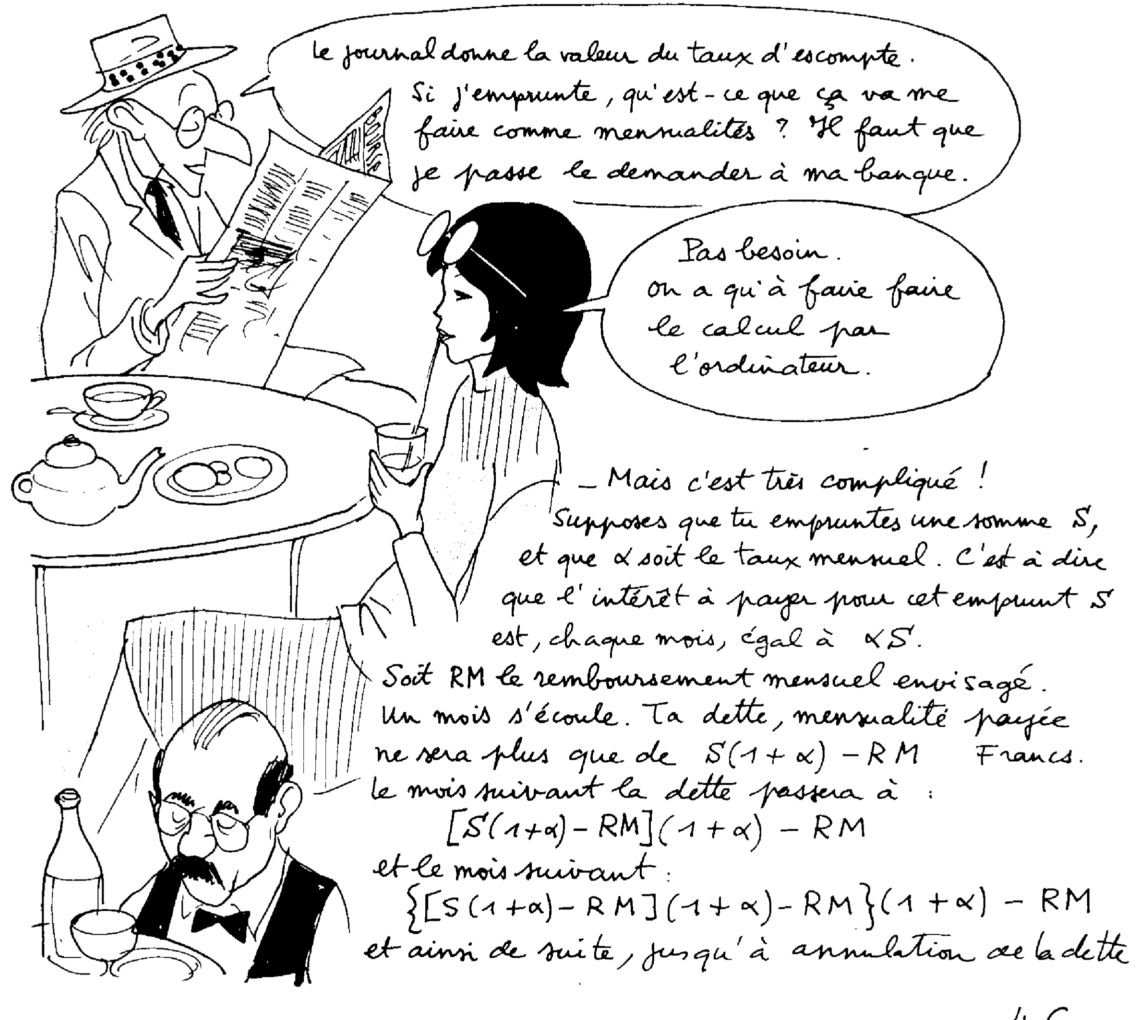
- Fouchtre ! tu as réussi à tout faire tenir sur une seule ligne.
- A gauche, tu as les deux TESTS, "supérieur à 989Ø" et "inférieur ou égal à 1Ø . Et, à droite, tu as les différentes actions qui en découlent, séparées par les deux points :
- Alors le THEN (ALORS) peut porter sur tout ce qui suit ?
- Oui, Ou tu mets un numéro de ligne (branchement conditionnel), ou tu mets la suite d'opérations à effectuer si le test est satisfait (*)

Alors le programme complet s'écrira :

```
1Ø      PRINT"CALCUL D'IMPOT SUR LE REVENU": PRINT
2Ø INPUT"DONNEZ VOTRE REVENU IMPOSABLE :";R : PRINT
3Ø INPUT"DONNEZ LE NOMBRE DE PARTS : ";N : PRINT
4Ø QF= R/N :PRINT"VOTRE QUOTIENT FAMILIAL EST : ";QF : PRINT
5Ø IF QF<=989Ø THEN T=Ø : I = Ø : GOTO 18Ø
6Ø IF QF>989Ø AND QF<=1Ø34Ø THEN T= 5 : I =R*Ø.Ø5-494.5*N: GOTO 18Ø
7Ø IF QF>1Ø34Ø AND QF<=1227Ø THEN T=1Ø : I=R * Ø.1 - 1Ø11.5 *N:GOTO 18Ø
8Ø IF QF>1227ØAND QF<=1941Ø THEN T=15 : I=R* Ø.15 - 1625 *N : GOTO 18Ø
9Ø IF QF>1941ØAND QF<= 2495Ø THEN T=2Ø : I=R * Ø.2Ø - 2595.5*N: GOTO 18Ø
1ØØ IF QF >2495ØAND QF<= 3136Ø THEN T=25 :I=R* Ø.25 - 3843*N :GOTO 18Ø
11Ø IF QF>3136ØAND QF<= 3794Ø THEN T=3Ø :I =R*Ø.3Ø - 5411 * N: GOTO 18Ø
12Ø IF QF>3794ØAND QF <= 4377Ø THEN T= 35 : I =R *Ø.35 - 73Ø8 *N : GOTO 18Ø
13Ø IF QF>4377ØAND QF <=7294Ø THEN T=4Ø : I = R *Ø.4 - 9496.5*N :GOTO 18Ø
14Ø IF QF>7294ØAND QF <=1ØØ32Ø THEN T=45: I = R * Ø.45 - 13143.5*N :GOTO 18Ø
15Ø IF QF>1ØØ32Ø AND QF<=11866Ø THEN T=5Ø : I = R * Ø.5 - 18159.5 *N :GOTO 18Ø
16Ø IF QF>11866Ø AND QF<=135ØØØ THEN T=55 : I=R* Ø.55- 24Ø92.5 *N :GOTO 18Ø
17Ø IF QF>135ØØØ THEN T= 6Ø : I= R*Ø.6 - 3Ø842.5 *N
18Ø PRINT"VOTRE TAUX D'IMPOSITION EST:";T;" POUR CENT" : PRINT
19Ø PRINT"MONTANT DE VOTRE IMPOT SUR LE REVENU:";I;" F"
2ØØ END
```

* On notera que les signes de l'arithmétique sont (addition +) (division /) (soustraction -) (multiplication *)

EMPRUNTER DE L'ARGENT



— Mais c'est très compliqué ! Supposes que tu empruntes une somme S, et que α soit le taux mensuel. C'est à dire que l'intérêt à payer pour cet emprunt S est, chaque mois, égal à αS.

Soit RM le remboursement mensuel envisagé. Un mois s'écoule. Ta dette, mensualité payée ne sera plus que de $S(1+\alpha) - RM$ Francs. le mois suivant la dette passera à :

$$[S(1+\alpha) - RM](1+\alpha) - RM$$

et le mois suivant :

$$\{[S(1+\alpha) - RM](1+\alpha) - RM\}(1+\alpha) - RM$$

et ainsi de suite, jusqu'à annulation de la dette

– Mais si ton emprunt tourne sur des dizaines de mois, ta formule devient plutôt longue !

– Attends ! la question que tu poses est : " si je rembourse en NA années quelle sera ma mensualité RM ?". De fait, ce calcul donne l'INVERSE : Quand on choisit le remboursement mensuel RM, il fournit par récursivité, le temps nécessaire pour éponger la dette. En se servant de cette "formule" il faudra donc faire un grand nombre de fois ce calcul, en essayant diverses valeurs de RM, jusqu'à ce qu'on tombe pile, ou suffisamment près.





On va commencer, bien sûr, par "ENTRER" tout ce qui est nécessaire.

```
10 [Effaçage Ecran]
20 INPUT "TAUX D'ESCOMPTE, EN %";TE
30 A = TE /1200 : PRINT
40 INPUT "SOMME EMPRUNTEE:";S : PRINT
50 INPUT "SUR COMBIEN D'ANNEES ?";NA
```

A est le TAUX MENSUEL. On a conjugué une division par cent (%) et par douze (les mois de l'année).

```
60 NM = NA*12 :REM NOMBRE DE MOIS
```

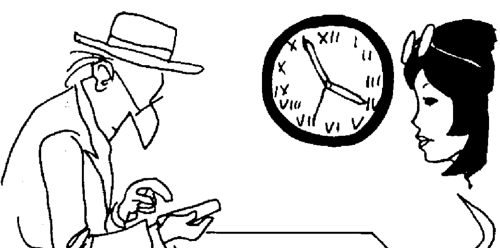
– Voilà. Plus qu'à résoudre le problème. Je serais curieux de voir comment tu vas t'y prendre.

– Exactement comme tu l'as défini. Il suffit de convertir en programme cette manière de procéder (récursive), cet ALGORITHME.

– Je suppose que tu vas prendre une valeur de départ pour RM.
– Tu remarqueras que RM a pour minimum αS. En effet, si RM = αS le temps de remboursement est.... infini, puisque tu ne fais que payer l'intérêt, mensuellement.
– Il faudrait commencer par RM = αS plus quelque chose. Mais quoi ? le plus simple est de fixer une précision PR sur le calcul de RM, par exemple 100F. Tu peux donc prendre comme RM initial la valeur $\alpha S + 100$.
– On remboursera alors, mois après mois, en INCRÉMENTANT.
– Mais sans dépasser le nombre de mois NM envisagé. Au bout de ce laps de temps, si la dette n'est pas épongée, c'est que le RM choisi était trop faible. On l'accroîtra donc de 100F, et ainsi de suite, jusqu'à ce que ça colle. Cela donnerait :

```
7Ø RM=S*A
8Ø RM= RM+1ØØ
9Ø M=Ø : S1=S : REM INTERMÉDIAIRE CALCUL
1ØØ S1=S1*(1+A)-RM : M=M+1
11Ø IF M<>NM THEN 1ØØ
12Ø IF S1>Ø THEN 8Ø
13Ø PRINT "REMBOURSEMENT MENSUEL : "; RM
```

ligne 7Ø on calcule la valeur minimale de RM
ligne 8Ø on incremente de 1ØØ (cent Francs)
Ligne 9Ø On commence un cycle. M est le nombre de mois, qui est mis à zéro. La dette sera S1, elle varie au cours du temps. On calcule S1 à la ligne 1ØØ. et on INCRÉMENTE M.
Ligne 11Ø si M est différent de NM, alors on passe un mois de plus (retour en 1ØØ). Sortie automatique de cette BOUCLE lorsque M = NM. On regarde alors (ligne 12Ø) si la dette subsiste. Si oui, on accroît RM (retour en ligne 8Ø), sinon on donne le résultat : ligne 13Ø



Consommant à la terrasse d'un café Achille composa le programme sur le petit ordinateur de poche de Sophie, qui se programmait en BASIC.

Dis-donc, on va y passer l'après-midi. Cela ne va pas si vite que cela !..

Ce n'est pas non plus la même machine, oncle Achille. Et puis on a pris un PAS DE CALCUL sans doute un peu faible !

– Oui, mais si on accroît le PAS, l'INCRÉMENT de RM, on va diminuer la précision du résultat !

– On va faire autrement. A la ligne 8Ø, au lieu de 1ØØ nous allons mettre par exemple 5ØØ. On va ainsi calculer le remboursement mensuel à 500 Francs près.

– Bon, ça va plus vite, et après ?

– Appelle R1 ce premier résultat. Tu sais alors que la solution exacte, au centime près, est entre R1 et R1-5ØØ. Tu prends donc R1-5ØØ comme nouvelle valeur de départ pour le remboursement mensuel RM. Et tu réédites l'opération avec, cette fois, un incrément de cinquante. Tu obtiendras R2, valeur approchée à cinquante Francs près. Démarres un troisième calcul avec R2-5Ø comme valeur initiale, avec un incrément de cinq. Tu auras une nouvelle valeur du remboursement mensuel à cinq francs près.

– Autrement dit, cette PROCÉDURE ITÉRATIVE m'amènera à un résultat précis en un temps cette fois acceptable ?

– Oui.

Cerise et Achille composèrent alors le programme ci-après :

RÉSOLUTION ITÉRATIVE :

```
10 [ Effaçage Ecran ]
20 INPUT "TAUX D'ESCOMPTE : "; TE
30 A = TE/1200 : PRINT
40 INPUT "SOMME EMPRUNTEE:"; S : PRINT
50 INPUT "NOMBRE D'ANNEES : "; NA
60 NM = NA*12 : REM NOMBRE DE MOIS
70 RM = S * A : REM VALEUR INITIALE
80 FOR I = 0 TO 2
90 IC = 500/(10^I) : REM INCREMENT RM
100 S1 = S : M = 0 : RM = RM + IC
110 S1 = S1 * (1+A) - RM : M = M+1
120 IF M<>NM THEN 110
130 IF S1 > 0 THEN 100
140 RM = RM - IC : REM UN PAS EN ARRIERE
150 NEXT I : RM = RM + IC
160 PRINT "MENSUALITES DE : " RM; " F"
170 END
```

RM

J'ai dépassé

un pas en arrière

Il y a trois RETOURS EN BOUCLE imbriqués.

Ligne 120 : retour en 110 : si le nombre de mois est inférieur au nombre imposé NM.

Ligne 130 : Si au nombre de mois NM la dette n'est pas épongée on incrémente RM de IC en 100 et on recommence.

La boucle 80-150 a trois cycles (0,1,2). L'incrément de RM vaudra successivement $500/10^0 = 500$; $500/10^1 = 50$; $500/10^2 = 5$ La précision du calcul, le PAS, prendra donc les valeurs successives 500, 50, 5 (Francs). A la ligne 140 on fait un pas en arrière, pour repartir d'une valeur par défaut, avec un incrément dix fois plus faible

Reprise de la Progression avec un PAS dix fois plus petit, etc...


Ce programme permet donc de calculer les mensualités à 5F près, pour n'importe quel emprunt, taux d'escompte et durée du prêt.
La version ordinateur de poche nous a donné le resultat en 33 minutes Sur ton micro ordinateur ce temps serait 50 fois plus faible.
Soit 40 secondes
_ Une autre question : quand on veut rembourser un emprunt en cours de route, on s'étonne souvent d'avoir "si peu remboursé".
_ L'ordinateur peut te donner la réponse immédiatement. Il suffit de relancer le calcul et de stopper au bout d'un certain nombre de mois, ou d'années, et de voir ce qui reste à payer.
_ Mais si je fais une étude sur un plan d'emprunt, et que je ne souhaite pas poser cette question à chaque fois ?
_ Nous allons envisager maintenant des PROGRAMMES A OPTIONS

NOTION DE MENU



On disait tout à l'heure que la programmation avait une parenté avec la cuisine. Ici, nous avons un menu à la carte. Si nous optons pour tel plat, cela va déclencher un certain nombre d'opérations en cuisine, qui aboutiront à la livraison du plat demandé.

– Imagine un restaurant qui ne va confectionner qu'un seul plat à la fois. Le travail accompli, le cuisinier reviendra te dire : « et ensuite ?... ». La suite pouvant éventuellement être : " merci, j'ai fini ".

– Ce restaurant devient un ordinateur ?!?

– Attends, regarde :

```
10 [Effaçage Ecran]
20 PRINT" VOULEZ-VOUS:":PRINT
30 PRINT "1-EMPRUNTER DE L'ARGENT":PRINT
40 PRINT" 2-PLACER DE L'ARGENT": PRINT
50 PRINT" 3-REMBOURSER EMPRUNT": PRINT
90 PRINT" 9-SORTIR DU PROGRAMME": PRINT
100 INPUT"VOTRE CHOIX: ";R
110 IF R=9 THEN END
120 IF R=1 GOSUB 1000
130 IF R=2 GOSUB 2000
  :        :      :
200 GOTO 10
```

Sur l'écran je verrai apparaître :



VOULEZ-VOUS :
1-EMPRUNTER DE L'ARGENT
2-PLACER DE L'ARGENT
3-REMBOURSER EMPRUNT
9-SORTIR DU PROGRAMME
VOTRE CHOIX ■

SOUS PROGRAMMES

– Si ta réponse est 1 l'instruction GOSUB 1ØØØ va t'envoyer à la ligne 1ØØØ.

– Comme un GOTO 1ØØØ ?

– Non. Si ce SOUS-PROGRAMME 1ØØØ se termine par une instruction spéciale RETURN, le retour sera assuré à la ligne qui suit immédiatement le GOSUB, c'est à dire la ligne 13Ø.

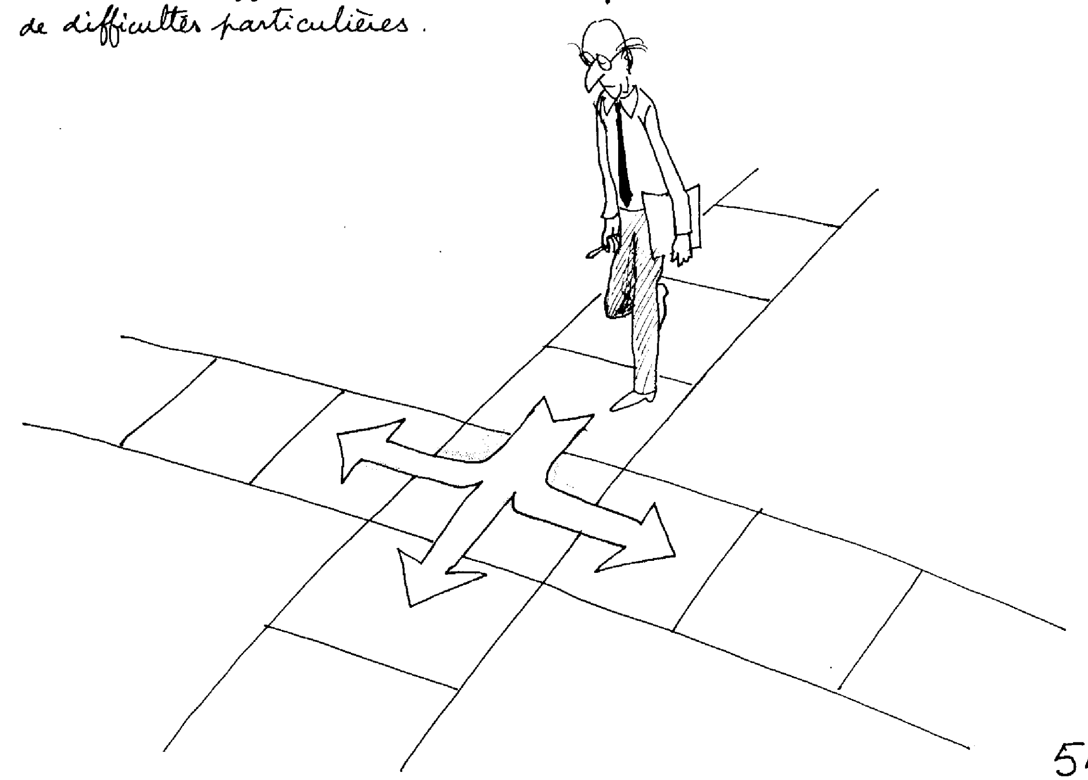
Il est commode de mettre ses sous-programmes à des adresses simples : 1ØØØ, 2ØØØ, 3ØØØ, etc... et de "bloquer" à cet effet les lignes 1ØØØ à 1999, 2ØØØ à 2999, 3ØØØ à 3999, etc... On mettra systématiquement le RETURN aux lignes 1999, 2999, 3999, etc...

– Bon, il suffit que je reprenne le programme de la page 50 et que je le renumérote :

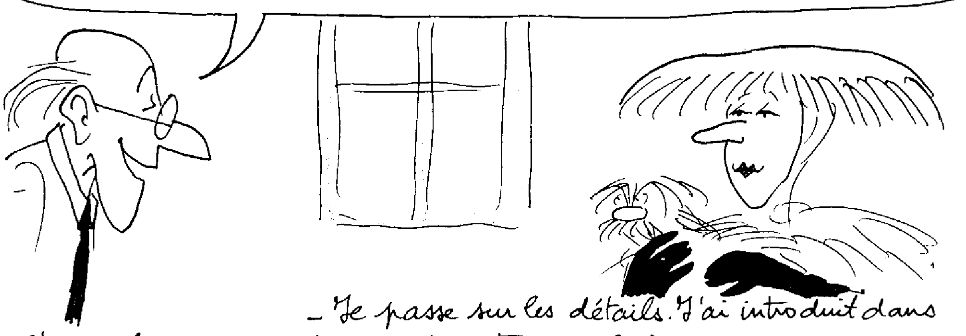
```
1Ø1Ø [Effaçage Ecran]
1Ø2Ø INPUT "TAUX D'ESCOMPTE";TE
1Ø3Ø A = TE/12ØØ : PRINT
      ETC... ETC...
1999 RETURN
```

– Pourquoi est-ce que les options sont numérotées 1, 2, 3, 9...?
– Ça te laisse la place pour en mettre d'autres.
– De même qu'il reste de la place entre 13∅ et 2∅∅, pour indiquer le sous programme correspondant. Cela n'a rien d'obligatoire, mais c'est commode d'associer l'option 4 à un sous-programme 4∅∅∅-4999 l'option 5 à un sous programme 5∅∅∅-5999, etc...

Moneyback compléta le programme présent dans l'ordinateur en introduisant effectivement les sous programmes 2 et 3, sans rencontrer de difficultés particulières.



Marie-Thérèse, je viens de faire un programme avec deux options. Dans la première, tu donnes les cinq premiers chiffres de ton numéro de sécurité sociale et l'ordinateur t'indique ton sexe, ton année de naissance, et ton mois. l'autre option réalise l'opération inverse



— Je passe sur les détails. J'ai introduit dans l'ordinateur un numéro de sécurité sociale N

— Par où, mon ami ?

— Euh... mais comme cela, à l'aide des touches.

— A l'aide des touches ?...

— Maintenant il s'agit d'isoler le premier caractère, à gauche. Si c'est un 1 le sexe est masculin. Si c'est un 2, il est féminin. J'ai trouvé dans le manuel une instruction INT qui donne la partie entière d'un nombre.

— La partie entière ?...

— Alors j'écris :

```
1010 [Effaçage Ecran]
1020 INPUT "VOTRE NUMERO ? "; N : PRINT
1030 S = N / 10000 : S = INT (S)
1040 IF S = 1 THEN PRINT "SEXE MASCULIN" : GOTO 1070
1050 IF S = 2 THEN PRINT "SEXE FEMININ" : GOTO 1070
1060 GOTO 1020
1070 REM AMBIGUITE LEVEE
```

– Et pour la suite ?
– Eh bien je connais le premier chiffre du numéro, qui en comporte cinq.
Je n'ai qu'à faire :

```
1080 A = N - S * 10000
```

Il ne me restera plus que les quatre nombres suivants. Si maintenant je divise par cent et que je prenne la partie entière :

```
1090 AN=INT(A/100)
```

J'aurai les deux chiffres qui indiquent l'année de naissance, et j'imprimerai :

```
1100 IF S=1 THEN PRINT"VOUS ETES NE EN 19";AN:PRINT
1110 IF S=2 THEN PRINT"VOUS ETES NEE EN 19";AN:PRINT
```

Puis je fais :

```
1120 B = A - AN * 100
1130 IF B = 1 THEN M$ = "JANVIER"
1140 IF B = 2 THEN M$ ="FEVRIER"
   :
1310 IF B=12 THEN M$="DECEMBRE"
1320 PRINT"AU MOIS DE ";M$

1999 RETURN
```

Sophie était arrivée sur ces entrefaites :

C'est très bien, oncle Achille (*)
L'informatique est une école
de débrouillardise

(*) Le lecteur complètera le programme en écrivant le sous programme de la 2e option et le "Menu"

TRAITEMENT DE TEXTE

- Mais on peut procéder autrement. Nous faisons ici une informatique CONVERSATIONNELLE où il y a échanges, dialogue, avec l'utilisateur à travers un système de questions - réponses. Mais l'ordinateur n'est qu'une machine. On ne peut pas dire qu'il comprenne ce qu'on lui dit. Il est donc essentiel d'éviter toute ambiguité. Imagine une question à laquelle il faut répondre par OUI ou par NON. Ecrivons par exemple :

```
510 INPUT "VOTRE REPONSE :"; R$
520 IF R$ = "OUI" THEN 110
530 IF R$ = "NON" THEN END
```

- Que se passera-t-il si l'utilisateur répond OK

Comme cette réponse n'est pas OUI l'ordinateur n'ira pas à la ligne 110.

Comme cette réponse R$ n'est pas NON, il ne s'arrêtera pas non plus. S'il existe en aval un sous programme quelconque, il l'exécutera comme un aveugle et finira par s'arrêter en affichant :

```
RETURN WITHOUT GOSUB
```


C'est à dire que ce sous programme aura été exécuté sans qu'il y ait eu APPEL de celui-ci par un ordre GOSUB, ce qui est illicite. On préfèrera écrire :

```
510 INPUT "VOTRE REPONSE:"; R$ : PRINT
520 IF R$ = "OUI" THEN 110
530 IF NOT(R$="NON") THEN 510
540 END
```

Imparable ! Si la réponse est OUI, branchement en ligne 110 (par exemple) Puis si la réponse N'EST PAS NON, reformulation de la question

– Mais revenons à cette histoire de numéro d'identification. Supposons qu'au lieu de le considérer comme un NOMBRE, on le considère comme une **CHAINE DE CARACTERES**. Et qu'on le **SAISISSE** à l'aide d'un INPUT N$

Primo, tu peux vérifier que l'utilisateur a bien donné cinq chiffres, ni plus, ni moins. Pour cela il suffit de compter le nombre de caractères de la chaine N$, ce qui se fait immédiatement à l'aide de LEN (*)

```
10 INPUT"DONNEZ LES CINQ PREMIERS CHIFFRES
DE VOTRE NO D'IDENTIFICATION :"; N$ : PRINT
20 L = LEN(N$)
30 IF L<5 OR L>5 THEN 10
```

Le sens de cette ligne 30 étant : si L est inferieur à 5 ou superieur à 5

On aura aussi pu mettre IF L<>5 (différent de) ou IF NOT(L=5) (si L n'est pas 5)

Ceci fait, l'instruction LEFT$ (left en anglais veut dire gauche) permet d'isoler un paquet de lettres en partant de la gauche, dans la chaine de caractères N$.

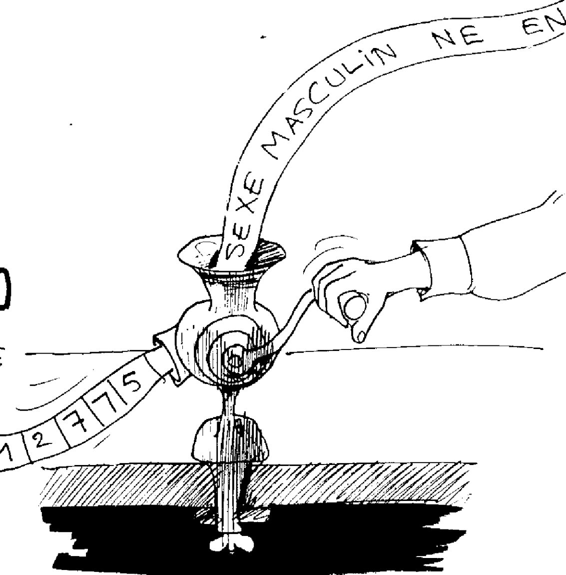
Il s'agit ici de prendre un caractère et on écrira :

```
40 S$ = LEFT$(N$,1)
```

S$ sera une **SOUS-CHAINE DE CARACTERES**

Puis on enchainera :

```
50 IF S$="1" THEN PRINT"SEXE
MASCULIN":PRINT : GOTO 90
70 IF NOT(S$="2") THEN 10
80 PRINT "SEXE FEMININ"
90 etc... etc..
```



* de LENGHT, en Anglais, longueur

Pour isoler le 2e et le 3e caractère, on utilisera une instruction MID$ (middle en anglais veut dire milieu).

```
9Ø AN$ = MID$(N$, 2, 2)
```

MID$ (N$, P, Q) veut dire : isoler Q caractères à partir du Pième (compris) en partant de la gauche, dans la chaine N$.
Ces groupes de lettres, ou de caractères, forment des SOUS-CHAINES.



– La communication avec l'ordinateur va devenir de plus en plus sophistiquée avec le temps. On cherche par exemple à mettre au point de (bonnes) machines à traduire. Il ne s'agit plus de traduire des mots, mais des phrases entières.
Le premier travail de l'ordinateur sera alors de ranger les mots d'une phrase, initialement séparés par un "blanc" dans la chaine P$. On parcourt celle-ci, caractère par caractère, et dès qu'on rencontre un blanc, on s'arrête et on crée une sous chaine qui sera le mot isolé.

Ecrivons : 1Ø [Effaçage Ecran]

```
2Ø PRINT"VOTRE  PHRASE :": PRINT
3Ø INPUT P$
4Ø P$ = P$ + " ":REM RAJOUTER  BLANC
5Ø I = Ø :REM MISE A ZERO COMPTEUR
6Ø I = I + 1 : REM  INCREMENTATION
```

```
7Ø C$ = MID$(P$,I,1)
```

Ceci isole le ième caractère de la chaine P$

```
8Ø IF C$<>" " THEN 6Ø
```

Si C$ n'est pas un blanc alors aller au caractère suivant.

```
9Ø M$ = LEFT$(P$,I-1)
```

le ième caractère a été identifié comme blanc.
Donc les (I-1) premiers caractères de P$ forment un MOT.

```
1ØØ PRINT M$ : L=LEN(P$)
11Ø IF L=I THEN END
```

si le blanc identifié est le dernier caractère de la chaine, alors le travail est terminé.

```
12Ø P$ = RIGHT$(P$,L-I) :GOTO 5Ø
```

Sinon on va amputer la chaine des I premiers caractères (le mot plus le blanc), garder la portion droite grâce à l'instruction RIGHT$ et revenir en 5Ø : nouvelle exploration de la chaine

Schématiquement :

CHAINE DE CARACTERES

CHAINE DE CARACTERES

I — tiens, un blanc!

M$ = CHAINE

I-1 caractères

CHAINE DE CARACTERES

I caractères — (L-I) caractères

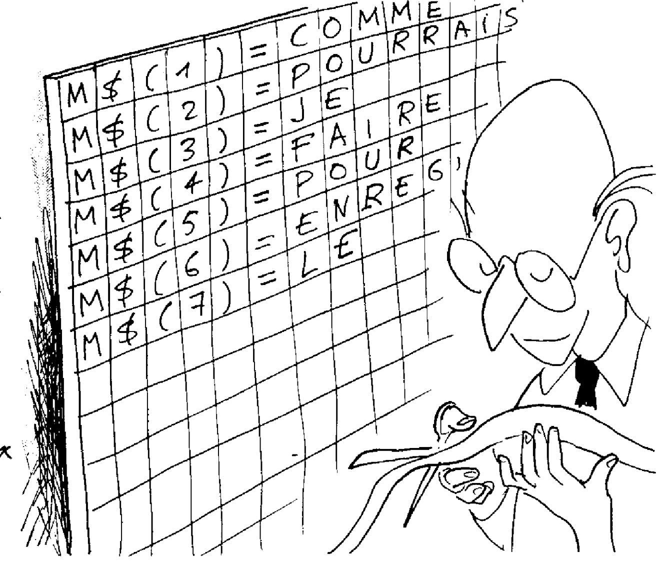
DE CARACTERES

I — tiens, un blanc!

VARIABLES INDICÉES

— Comment pourrais-je faire, dit Achille, pour enregistrer le "premier" mot, puis le "second" etc... Il faudrait un chiffre, un INDICE, qui permettrait de se repérer. Par exemple l'indice J. On pourrait parler du $J^{ème}$ mot de la phrase.

— Pas de problème, c'est un langage que l'ordinateur comprend très bien. Tu peux manipuler **M$(J)**



Moneyback écrivit donc :

```
10 [Effaçage écran]
20 PRINT "VOTRE PHRASE :" : PRINT
30 INPUT P$   : PRINT
40 P$ = P$ + " " :REM RAJOUTER UN BLANC A DROITE
50 J = 0 :REM MISE A ZERO COMPTEUR MOTS
60 J = J + 1 : REM INCREMENTATION
70 I = 0 :REM MISE A ZERO COMPTEUR CARACTÈRES
80 I = I + 1 : REM INCREMENTATION
90 C$ = MID$(P$, I, 1) :REM ISOLER CARACTERE
100 IF C$<>" " THEN 80
```

```
110 M$(J)=LEFT$(P$,I-1):REM ISOLER LE MOT
120 PRINT M$(J):PRINT
130 L=LEN(P$) :REM LONGUEUR CHAINE P$
140 IF L=I THEN END :REM FIN DE TACHE
150 P$=RIGHT$(P$,L-I):REM TRONQUER CHAINE
160 GOTO 60 :REM RECHERCHE DU MOT SUIVANT
```

Cette séquence a découpé la phrase en mots M$(J). On peut les réafficher en écrivant par exemple :

```
170 FOR K=1 TO J
180 PRINT M$(K):PRINT
190 NEXT K
```

– Mais alors, l'indice, c'est J ou c'est K ??

J contient, à la fin du découpage, le nombre de mots de la phrase. Si tu veux, tu pourrais écrire :

```
165 NM=J :REM NB DE MOTS
170 FOR J=1 TO NM
180 PRINT N$(J):PRINT
190 NEXT J
```

Comme cela tu peux réutiliser ta lettre J. Mais, pour l'ordinateur, ce qui compte, c'est la quantité numérique qui se trouve dans ces registres J ou K.

On a créé dans la mémoire une TABLE à un indice M$(1), M$(2), M$(3) etc... et c'est dans cette table que l'on puise, que cela soit avec un indice J ou K. Seul compte le NOMBRE contenu dans ces registres.

```
LES PANTOUFLES VOLAINT DANS L'AZUR FAMELIQUE
LES
PANTOUFLES
VOLAIENT
DANS
L'AZUR
FAMELIQUE
```

voilà ce que cela donne

62

ÉLÉMENTS DE GESTION




Mais oui ... enfin, je me comprends. Tout est clair, ou presque clair. c'est la VOIE, le TAO de l'informatique.
Je suis initié. Je vais pouvoir épater mes amis, mystifier mon personnel.
BASIC
PASCAL
FORTRAN
LOGO
Moneyback travailla une partie de la nuit, avec fièvre... Consultant fréquemment son manuel et pianotant nerveusement sur le clavier de la machine.

FICHIERS




```
1010 REM LISTE CLIENTS + COMPTES
1020 [Effaçage Ecran]
1030 I = 0
1040 I = I+1
1050 INPUT "NOM CLIENT "; N$(I) : PRINT
1060 PRINT "SON ADRESSE :" ; PRINT
1070 INPUT "RUE, NUMERO "; R$ (I)
1080 INPUT "ARRONDISSEMENT : "; AR $(I)
1090 INPUT "VILLE:" ; V$ (I)
1100 INPUT " PAYS : "; P$ (I) : PRINT
1105 INPUT "COMPTE = "; CC (I) : PRINT
1110 PRINT "IL S'AGIT BIEN DE : " : PRINT
1120 PRINT N$ (I)
1130 PRINT R$ (I)
1140 PRINT V$ (I); " "; AR$ (I)
1150 PRINT P$ (I)
1160 PRINT "COMPTE ="; CC (I); " F" : PRINT
1170 INPUT "CORRECT ?"; R$ : PRINT
1180 IF R$ ="OUI" THEN 1210
1190 IF NOT (R$ ="NON") THEN PRINT
"REPONDEZ PAR OUI OU NON" : GOTO 1170
1200 GOTO 1050
```

```
1210 PRINT: INPUT"UN AUTRE CLIENT?";R$: PRINT
1220 IF R$="OUI" THEN 1040
1230 IF NOT (R$="NON") THEN PRINT"RÉPONDEZ OUI OU NON":
 GOTO 1210
1240 NC=I : REM NOMBRE DE CLIENTS
```

— Oui, c'est pas mal. Tu crées des fichiers avec un indice de base, qui est le numéro du client. A partir de ce numéro tu pourrais aisément connaître :

- N$(I) le nom du client
- R$(I) sa rue et son numéro
- AR$(I) son arrondissement
- V$(I) la ville où il réside
- P$(I) son pays
- CC(I) l'état de son compte (avoir ou crédit)

— Je devrais pouvoir inclure ce programme dans un programme principal, un **MENU**, regroupant les différentes tâches.
— Peux tu les décrire ?
— Pour commencer, après avoir CRÉÉ le fichier, il faut pouvoir le COMPLETER, puis le CONSULTER, METTRE A JOUR LES COMPTES DES CLIENTS (au fur et à mesure des facturations ou des règlements de factures). Il faut aussi pouvoir RELANCER LES IMPAYES, ANALYSER LES COMPTES...
— Et bien d'autres choses encore. Pour le moment, disons-nous que cela suffira. A titre d'exercice.
— Nous allons écrire le MENU ?
— Oui ...

– Combien as tu de clients ?
– De gros clients, une cinquantaine...
– On va prendre cent. L'indice pourra aller jusqu'à cent. Il faut alors RESERVER DES PLACES EN MEMOIRE.
– Que veux-tu dire ?
– Pour les variables indicées, tu disposes automatiquement de dix places correspondant aux valeurs d'indice : (Ø, 1, 2, 3, 4, 5, 6, 7, 8, 9).
Si la valeur de l'indice peut dépasser 9, comme c'est le cas ici, il faut le spécifier à l'aide d'ordres DIM (pour dimension)
Supposes que tu veuilles créer des tables, des fichiers N$(I), R$(I) AR$(I), V$(I), P$(I), CC(I).

On mettra : DIM N$(1ØØ), R$(1ØØ), AR$(1ØØ), V$(1ØØ), P$(1ØØ), CC(1ØØ)

Cette ligne réservera cent places (en fait cent dix car l'indice zéro compte) pour chacun des fichiers.

```
10 REM  PROGRAMME DE GESTION DE COMPTES-CLIENTS
20 REM  RESERVATION DE PLACES EN MEMOIRE :
30 DIM N$(100),R$(100),AR$(100),V$(100),P$(100),CC(100)
100 [Effaçage Ecran]

110 REM  PROGRAMME-MENU :
120 PRINT"VOULEZ-VOUS :":PRINT
130 PRINT"1-CREER UN FICHIER-CLIENTS": PRINT
140 PRINT"2-COMPLETER UN FICHIER-CLIENTS":PRINT
150 PRINT"3-CONSULTER UN FICHIER-CLIENTS":PRINT
160 PRINT"4-METTRE A JOUR UN COMPTE-CLIENT":PRINT
170 PRINT"5-RELANCER LES IMPAYES" : PRINT
180 PRINT"6-ANALYSER COMPTES CLIENTS":PRINT
190 PRINT"9-SORTIR DU PROGRAMME":PRINT
200 INPUT"VOTRE CHOIX :"; C : PRINT
210 IF C = 9 THEN END
220 IF C = 1 THEN GOSUB 1000
230 IF C = 2 THEN GOSUB 2000
240 IF C = 3 THEN GOSUB 3000
250 IF C = 4 THEN GOSUB 4000
260 IF C = 5 THEN GOSUB 5000
270 IF C = 6 THEN GOSUB 6000
300 GOTO 100
310 REM RETOUR AU MENU
```

Sur l'écran !

```
VOULEZ-VOUS :
1-CREER UN FICHIER-CLIENTS
2-COMPLETER UN FICHIER-CLIENTS
3-CONSULTER UN FICHIER-CLIENTS
4-METTRE A JOUR UN COMPTE-CLIENT
5-RELANCER LES IMPAYES
6-ANALYSER COMPTES CLIENTS
9-SORTIT DU PROGRAMME

VOTRE CHOIX? ■
```

– Bon, j'ai déjà écrit le sous-programme 1000.
– Mais, que fais-tu si tu te trompes ? si tu fais une erreur de frappe ?
– Ce serait très ennuyeux...
– On va mettre un petit circuit de vérification avant stockage définitif, qui limitera considérablement les erreurs.

```
1000 REM CONSTITUTION FICHIER-CLIENTS
1010 [Effaçage Ecran]
1020 REM MISE A ZERO DU NUMERO CLIENT:
1030 I=0
1040 I=I+1: REM INCREMENTATION
1045 IF I>100 THEN PRINT"DEPASSEMENT DE CAPACITE-MEMOIRE":
GOSUB 20000 : GOTO 1999
```

– Qu'est-ce qu'il y aura à cette adresse 20.000 ?
– Un petit sous-programme qui comportera une boucle de temporisation. Puis on renvoit en 1999 RETURN. Inutile de poursuivre la tâche puisqu'elle est considérée comme impossible. On saute donc en fin de sous-programme

```
1050 INPUT"NOM DU CLIENT :";N$(I) : PRINT
1060 PRINT"SON ADRESSE : " : PRINT
1070 INPUT" RUE : "; R$ :PRINT :REM DONNEE PROVISOIRE
1080 INPUT"NUMERO DANS LA RUE : "; N$ : PRINT
1087 R$(I) = N$ + " RUE " + R$
```

– Là nous faisons une CONCATENATION. CATENA veut dire chaîne en latin. Littéralement "mettre en chaîne". C'est l'opération inverse des troncatures MID$ ou LEFT$ ou RIGHT$

L'ORDINATEUR NE CONFOND PAS R$ et R$(I), CONSIDÉRÉS COMME DISTINCTS

Tu as un certain nombre de chaines de caractères (ici N$. "RUE" et R$, et l'aide du + tu formes une chaine par association de ces trois.

```
1090 INPUT"VILLE:";V$(I) : PRINT : INPUT"ARRONDISSEMENT:";AR$(I) : PRINT
1100 INPUT"PAYS:";P$(I) : PRINT
1105 INPUT"COMPTE:";CC(I) : PRINT
1110 PRINT"IL S'AGIT BIEN DE :":PRINT
1120 PRINT N$(I)
1130 PRINT R$(I)
1140 PRINT V$(I);"  ";AR$(I)
1150 PRINT P$(I)
1160 PRINT "COMPTE=";CC(I);" F" :PRINT
1170 INPUT"EST-CE CORRECT ?";R$ :PRINT
1180 IF R$="OUI"THEN 1210
1190 IF NOT(R$="NON")THEN PRINT "REPONDEZ PAR OUI OU PAR
NON" : GOTO 1170
1200 GOTO 1050 :REM SI ERREUR
1210 INPUT"UN AUTRE CLIENT?";R$ : PRINT
1220 IF R$="OUI"THEN 1040
1230 IF NOT(R$="NON")THEN PRINT
"REPONDEZ OUI OU NON" :GOTO 1210
1240 NC = I : REM NB DE CLIENTS
1999 RETURN
```

– Ce petit système permet de se rattraper en cas d'erreur. On repart alors en ligne 1050 c'est à dire qu'on recommence

```
NOM DU CLIENT : LAMBERT
SON ADRESSE :
RUE : GRANET
NUMERO DANS LA RUE : 6
VILLE : PARIS
ARRONDISSEMENT : 7
PAYS : FRANCE
COMPTE : 16322
IL S'AGIT BIEN DE
LAMBERT
6 RUE GRANET
PARIS 7
FRANCE
COMPTE = 16322 F
EST-CE CORRECT?■
```

à entrer les données, mais avec le même indice I. Il s'agit toujours du même client. Par contre à la ligne 122ø, si la réponse est OUI on renvoit à la ligne 1ø4ø où on incrémente I, c'est à dire qu'on passe au client suivant.

Ce programme de création de fichier-clients ne va te servir qu'une fois. Ultérieurement tu te contenteras de le compléter, selon l'option 2. On pourrait s'arranger même pour que l'option 1, après usage, disparaisse du menu.

— Parce que si je fais l'erreur de m'orienter vers cette option le premier ordre met à zéro le numero-client. Tout est fichu en l'air...

— Tu vois donc qu'il y a un gros problème de fiabilité. Il faut avoir pensé à tous les cas possibles, à toutes les erreurs.



le sous programme 2000 est dérivé du précédent.

```
2000REM COMPLETER FICHIER
2010 [Effaçage Ecran]
2020 PRINT"CE FICHIER CONTIENT :";NC;" CLIENTS" : PRINT
2030 NC=NC+1 : REM INCREMENTER NC
2040 PRINT"NOUVEAU CLIENT:" : PRINT
2050 INPUT"NOM : ";N$(NC) : PRINT
2060 PRINT"SON ADRESSE :"; PRINT
2070 INPUT"RUE:";R$(NC) : PRINT
2075 INPUT"NUMERO DANS LA RUE";N$: PRINT
2080 R$(NC)=N$+" RUE "+R$
2085 INPUT"VILLE:";V$(NC) : PRINT
2090 INPUT"ARRONDISSEMENT:";AR$ :PRINT
2100 INPUT"PAYS"; P$(NC): PRINT
2105 INPUT"COMPTE =";CC(NC) : PRINT
2110 PRINT"IL S'AGIT BIEN DE:":PRINT
2120 PRINT N$(NC)
2130 PRINT R$ (NC)
2140 PRINT V$(NC);" "; AR$(NC)
2150 PRINT P$ (NC)
2160 PRINT"COMPTE =";CC(NC);" F": PRINT
2170 INPUT"CORRECT ?";R$ : PRINT
2180 IF R$="OUI"THEN 2210
2190 IF NOT (R$="NON")THEN PRINT"REPONDEZ OUI OU NON" :
PRINT : GOTO 2170
2200 GOTO 2050
2210 INPUT"UN AUTRE CLIENT?";R$ : PRINT
```

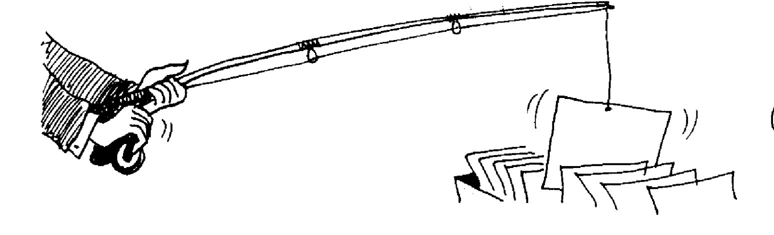
```
2220  IF R$ = "OUI" THEN 2040
2230  IF NOT (R$ = "NON") THEN PRINT "REPONDEZ OUI OU NON": PRINT: GOTO 2210
2999 RETURN
```

– Passons à la consultation du fichier :

```
3000 REM  CONSULTATION FICHIER-CLIENTS
3010 [Effaçage Ecran]
3020 PRINT " VOULEZ-VOUS :" : PRINT
3030 PRINT "1- EXAMINER UN COMPTE CONNAISSANT LE" : PRINT :
PRINT "NOM DU CLIENT" : PRINT : PRINT
3040 PRINT "2- IDEM EN CONNAISSANT SON NUMERO" : PRINT
3050 INPUT "VOTRE CHOIX ?" ; C
3060 IF C = 1 THEN GOSUB 10000
3070 IF C = 2 THEN GOSUB 11000
3999 RETURN
```

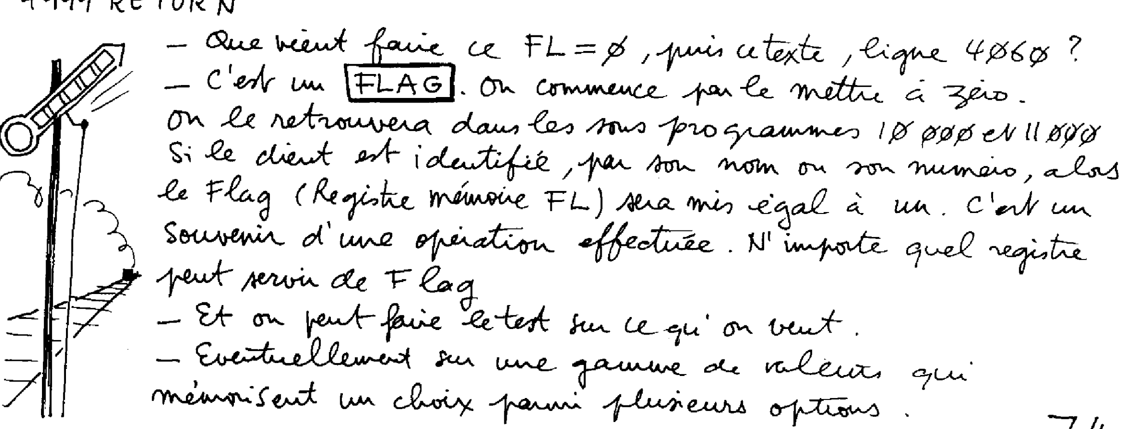
– Il y a un deuxième MENU !?!

– oui, c'est un SOUS-MENU. Et il envoit à des SOUS-OPTIONS et à deux autres sous-programmes (*) que nous préciserons plus loin. Il vont correspondre à l'identification du client dans le fichier.



(*) On devrait dire à des sous-sous-programmes...

```
4000 REM MISE A JOUR COMPTE-CLIENT
4010 [Effaçage Ecran]
4020 PRINT"1-VOUS CONNAISSEZ LE NOM DU CLIENT":PRINT
4030 PRINT"2-VOUS CONNAISSEZ SON NUMERO":PRINT
4040 FL=0: INPUT"VOTRE CHOIX ?";C :PRINT
4050 IF C=1 THEN GOSUB 10000
4055 IF C=2 THEN GOSUB 11000
4060 IF FL=0 THEN PRINT"ERREUR":GOSUB 20000:GOTO 4999
4070 [Effaçage Ecran]
4080 PRINT"IL S'AGIT BIEN DE :";N$(I) :PRINT
4090 PRINT"COMPTE=";CC(I);" F":PRINT
4100 PRINT"INDIQUEZ L'OPERATION, POUR UN REGLEMENT":PRINT
4110 PRINT"FAITES PRECEDER D'UN +":PRINT
4120 PRINT"POUR UN CREDIT, D'UN -":PRINT
4130 INPUT"MONTANT OPERATION:";OP :PRINT
4140 CC(I)=CC(I)+OP
4150 PRINT"NOUVEL ETAT DE COMPTE :";CC(I);" F":GOSUB 20000
4999 RETURN
```



— Que vient faire ce FL=0, puis ce texte, ligne 4060 ?

— C'est un FLAG. On commence par le mettre à zéro. On le retrouvera dans les sous programmes 10 000 et 11 000. Si le client est identifié, par son nom ou son numéro, alors le Flag (Registre mémoire FL) sera mis égal à un. C'est un souvenir d'une opération effectuée. N'importe quel registre peut servir de Flag

— Et on peut faire le test sur ce qu'on veut.

— Eventuellement sur une gamme de valeurs qui mémorisent un choix parmi plusieurs options.

– Et pour relancer mes impayés ?

```
5000 REM   RELANCE   IMPAYES
5010 [ Effaçage Ecran ]
5020 PRINT "SONT DEBITEURS :" : PRINT
5025 FOR I = 1 TO NC : REM EXPLORATION FICHIER
5030 IF CC(I)<0 THEN PRINT N$(I),P$(I),CC(I) : PRINT
```

– c'est un système (la virgule) qui permet d'afficher plusieurs renseignements sur une même ligne. Ici le nom, le pays, la somme due.

```
5040 NEXT I
5045 PRINT
5050 INPUT "APPUYEZ SUR LA TOUCHE (FIN DE MESSAGE)"; B$
5999 RETURN
```

– Cette ligne 5045 met l'ordinateur en attente d'une donnée, d'une chaine de caractères, qu'il mettra dans un registre B$. Si on appuie directement sur la touche [Fin de message] une chaine "VIDE" sera stockée.

– Pourquoi faire ???

– Rien ... c'est une des multiples astuces pour stopper l'ordinateur (pour noter ce qu'il fournit). Il repartira sur une simple pression de la touche [Fin de message]

– Commode, en effet.

```
6000 REM ANALYSE (SOMMAIRE) DES COMPTES-CLIENTS
6010 [Effaçage Ecran] : A = 0 : P = 0
6020 PRINT "NOMBRE DE CLIENTS :"; NC : PRINT
6030 FOR I = 1 TO NC : REM EXPLORATION FICHIER
6040 IF CC(I) > 0 THEN A = A + CC(I)
```

... si le compte CC(I) du I^{eme} client est en positif, on le somme sur l'ACTIF A

```
6050 IF CC(I) < 0 THEN P = P + CC(I)
```

... en négatif, sommation sur le PASSIF

```
6060 NEXT I
6070 PRINT : PRINT "ACTIF ="; A ; " F" : PRINT
6080 PRINT : PRINT "PASSIF ="; P ; " F"
6090 GOSUB 20000 : REM TEMPORISATION
6999 RETURN
```


— Il nous reste à écrire les sous programmes 10000, 11000 et 20000

```
10000 REM EXAMEN DE COMPTE, NOMINATIF
10010 [Effaçage Ecran]
10020 INPUT "NOM DU CLIENT :"; N$ : PRINT
10030 I = 0
10040 I = I + 1 : REM INCREMENTATION
10050 IF N$ = N$(I) THEN FL = 1 : PRINT "COMPTE ="; CC(I)
; " F" : GOSUB 20000 : GOTO 10999
10060 I=NC THEN PRINT "NOM HORS FICHIER" : GOSUB 20000 : GOTO 10999
1070 GOTO 10040
10999 RETURN
```

Sophie commenta :

– ligne 1ØØ5Ø si le client est reconnu, le Flag FL est mis à 1, et on indique son état de compte. Sortie de boucle. Le registre I mémorise le numéro du client.

```
11ØØØ REM CONSULTATION COMPTE PAR NUMERO
11Ø1Ø [Effaçage Ecran]
11Ø2Ø INPUT"NUMERO DU CLIENT?"; I: PRINT
11Ø3Ø IF  >NC THEN PRINT"ERREUR,NUMERO TROP GRAND":
 GOSUB 2ØØØØ : GOTO 11999
11Ø4Ø PRINT"NOM DU CLIENT:"; N$(I) : PRINT
11Ø45 PRINT"COMPTE:";CC(I); " F": GOSUB 2ØØØØ :FL= 1
11999 RETURN
```

Encore une fois le Flag indique si le client a été trouvé et le registre I contient son numéro, correct ou non.

– Reste la boucle de temporisation :

```
2ØØØØ REM  TEMPORISATION
2ØØØ5 FOR T= 1 TO 2ØØØ : NEXT T
2Ø999 RETURN
```



– Tu mets toute la boucle sur une seule ligne ?

– Aucun problème...

c'est déjà un sacré programme...

— Un petit début, oncle Achille. Mais tout cela t'a quand même appris les grandes lignes de la programmation. Il faudrait encore beaucoup de choses pour compléter ce programme. Par exemple introduire quelquechose pour transférer les fichiers sur disques.

— Mais... je n'ai qu'à faire SAVE ?

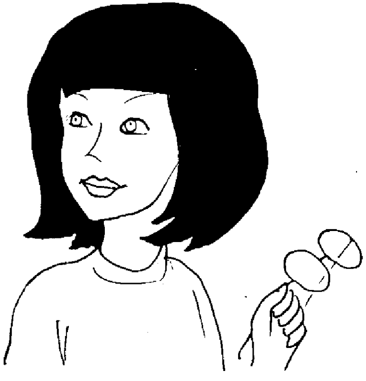
— Ce faisant, tu stockes le PROGRAMME, mais pas les contenus de mémoires.

— Ah... bon !

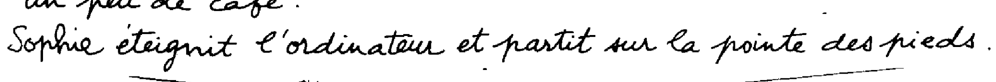
— Et rappelle-toi que lorsque tu fais RUN, la première chose qui se passe, c'est la remise à zéro de toutes les mémoires.

— Alors j'ai travaillé pour rien !

— Non, il manque certaines choses à ce programme. Mais ceci est une autre histoire...



La journée s'annonçait belle. Marie-Thérèse alla préparer



FIN

INSTRUCTIONS PRINCIPALES DU BASIC

On a indiqué, dans la colonne de gauche, les pages où ces instructions ont été introduites, au fur et à mesure, et décrites.

13	C=3 ou A=-8	Indique que les valeurs 3 et moins 8 sont inscrites dans les registres C et A
14	PRINT B ?B	lecture du contenu d'un registre numérique B.
14	ENTER RETURN	Instruction de fin de message.
14	CLEAR	met à zéro tous les contenus des registres, numériques ou littéraux.
15	SYNTAX ERROR	Message d'erreur le plus courant
16	N$="SOPHIE"	création d'un registre N$ contenant la chaîne de caractères CERISE
22	END	Instruction d'arrêt
23	HOME	Pour effacer l'écran
23	1 PRINT "CLAVIER EST UN ANE"	ligne programme BASIC
24	NEW	Efface le programme
27	REM Article CP90	Permet d'inscrire des notes dans un programme.
27	LIST	déclenche l'affichage de tout programme présent en mémoire.
27	LIST 3Ø	Affiche seulement la ligne 3Ø.
27	LIST 5, 8Ø	Affiche les lignes 5 à 8Ø, incluses.
28	DEL 5Ø,11Ø	Efface les lignes 5Ø à 11Ø, incluses
3Ø	INPUT N	Exécution du programme interrompue pour attente d'une donnée numérique.
3Ø	INPUT N$	Même chose avec attente d'une chaîne de caractères.
3Ø	INPUT "DATE LETTRE CLIENT:"; D$	Idem avec questionnement.

31	14Ø PRINT "QUANT À L'ARTICLE"; AR$; " NOUS NE" : PRINT	
	Façon d'afficher plusieurs chaines sur l'écran, bout à bout.	
34	SAVE / LOAD / CSAVE / CLOAD	recopie ou charge (sur, à partir de) la mémoire externe.
37	N=N+1	Incrémentation dans le registre N, l'incrément étant de 1.
38	5Ø GOTO 3Ø	branchement inconditionnel à la ligne 3Ø
4Ø	5Ø IF N=15Ø THEN 7Ø	Branchement conditionnel (avec Test) vers ligne 7Ø
41	2Ø FOR N=2 TO NM	Début de boucle FOR NEXT (de N=2 à la valeur NM).
41	4Ø NEXT N	Renvoit à la ligne d'initiation de la boucle.
53	GOSUB 1ØØØ	Envoit vers le sous programme débutant à la ligne 1ØØØ
54	1999 RETURN	indique une fin de sous programme.
56	AN=INT(A)	AN= partie entière de A.
58	L=LEN(N$)	L est le nombre de caractères de la chaine N$
59	AN$=MID$(N$,2,2)	création de la sous-chaine AN$ à partir de N$
6Ø	M$=LEFT$(P$,5)	Idem. les 5 caractères, en partant de la gauche de P$.
6Ø	P$=RIGHT$(P$,4)	Idem. les quatre caractères, en partant de la droite de P$.
61	M$(J) / N(I)	Variables indicées.
65	IF NOT(R$="NON") THEN 11Ø	Test avec branchement sur réponse négative.
67	DIM N$(4ØØ), CC(4ØØ)	Réservations de place mémoire pour fichiers.

La suite de cet ouvrage initiera le lecteur au stockage des **DONNÉES** et à la manipulation de **FICHIERS**, d'abord intégrés dans le programme lui-même, sous forme de **DATA**, puis dans une mémoire externe.
Ce livre a été écrit au début des années quatre vingt, à une époque où certains micro-ordinateurs possédaient encore des mémoires externes à bandes, sous forme de cassettes, ou de mini-cassettes.

J'ai manqué de temps pour remettre cette partie de l'ouvrage au goût du jour.

Le programme **LOGOTRON**, qui engendre aléatoirement des **NÉOLOGISMES** est né à la fin des années soixante-dix. Nous l'avons présenté à la télévision (TF1), en déclarant "que nous avions découvert, à "l'institut de Logotronique de la faculté des lettres d'Aix en Provence, que nous étions à la surface d'un gisement sémantique pratiquement inexploré.
Le Logotron peut engendrer 90.000 mots, dont la plupart n'existent pas dans le dictionnaire...


qui aurait imaginé qu'il puisse exister tant de mots
SCHKRRRRRTTTTT



SOCIE
MONEYBA
Salut !
Eh bien, vous en faites une tête !?
qu'est-ce qui se passe ?
Il y a que madame Pinardel, monsieur Lemur et moi, sommes en BRAINSTORMING
Nous faisons de la créativité de groupe
La situation est critique. les japonais nous talonnent, les suédois aussi.
Jusqu'ici nous étions les seuls sur le marché à produire nos cache-cache pots. Mais il nous en arrive maintenant de Hong-Kong et de Taïwan...

Avec leur main d'œuvre locale, ils nous passent devant. Nous ne sommes plus compétitifs

VENTES

Bien sûr, nous restons les seuls à produire nos fameux sachets pour étuis. Mais, pour combien de temps encore ?.. Il y aura un jour des contrefaçons asiatiques

Le péril jaune !

Pour prévenir un tel danger, une seule solution : INNOVER

Vous entendez, madame Pinardel, **INNOVER** !

innover, oui, monsieur le directeur.

Passons au **DEBRIEFING**. Voyons les idées de ce cher monsieur Lemur

Je pourrais peut-être aller faire un stage d'innovation quelquepart ?

un stage d'innovation !?! Et puis quoi encore ?

L'innovation, il faut avoir cela en soi.

Je me demande si on ne pourrait pas se servir de l'ordinateur ?..

L'ordinateur, madame Pinardel, mais ça n'est pas la potion magique ! Il ne peut faire le travail à votre place !

Mais il peut peut-être vous aider à innover

A innover !? mais, grands dieux comment ?

Suffit-il de lui dire : invente !?!

– Non, répondit Sophie. Mais il peut combiner des choses entre elles, de manière imprévisible. Et il peut en sortir une idée à laquelle personne n'aurait songé. A nous de faire le tri !..

– Je ne comprends pas !

– Prenons par exemple des mots comme BAROMÈTRE, TÉLÉPHONE. Ils sont construits à l'aide des RACINES :

BARO : pression
MÈTRE : mesure
TÉLÉ : loin
PHONE : son

BARO MÈTRE

Nous allons prendre tout un lot de mots décomposables de cette manière, et nous allons chercher d'autres combinaisons des racines, de manière à créer des **NÉOLOGISMES.**

Hum, ça ne veut pas dire grand'chose...

moi, j'ai plus de chance

BARO PHONE

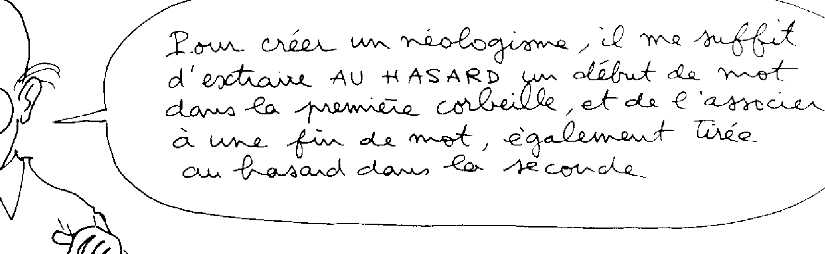
TÉLÉ MÈTRE

– Ah oui, fit remarquer madame Pinardel. TÉLÉ = LOIN, plus MÈTRE : mesurer. C'est un appareil... qui mesure au loin.
– Mais, chère madame, le télémètre, ça existe déjà !
– Bien sûr, oncle Achille. Mais, en poursuivant cette démarche, nous allons sûrement tomber sur des choses qui n'existent pas.
– Et alors, nous pourrons les .... **INVENTER !?!**
– Pourquoi pas? Si le TÉLÉMÈTRE n'avait pas existé, nous aurions pu chercher à quoi cela aurait bien pu servir, comment le concevoir, le fabriquer ...
– Et comment le VENDRE ! Cerise, ton projet m'intéresse ! Bon, au travail ! Madame Pinardel, allez me chercher le dictionnaire ...

Bon. J'ai coupé des tas de mots. A gauche les débuts de mots, les **PRÉFIXES**, et à droite les fins de mots, les **SUFFIXES**

— Madame Pinardel, allez me chercher deux corbeilles à papier
— Oui, monsieur le directeur.

Achille plaça les débuts de mots, en vrac, dans la première, et les fins de mots, également en vrac, dans la seconde



Moneyback se gratta le nez.

— Il me semble que je dois pouvoir disposer, dans la mémoire centrale de l'ordinateur de deux FICHIERS D$(I) et F$(J).

— Exact, commenta Sophie. Le signe $ indique que ces fichiers vont contenir des CHAINES DE CARACTÈRES. Les lettres I et J figurent les INDICES DE REPÉRAGE.

Madame Pinardel se plaignit de ne plus comprendre :
— Ah bon, ce sont des mathématiques !...
— Meunon, vous voyez des mathématiques partout. Prenons un lot de quelques débuts de mots :
CLEPTO, CHRONO, EPISTEMO, LOGO, NEO, PAN, PATHO

TABLEAUX

Sophie poursuivit :

– Je vais numéroter ces objets, ces CHAINES DE CARACTÈRES :

1 CLEPTO
2 CHRONO
3 EPISTEMO
4 LOGO
5 NEO
6 PAN
7 PATHO

Et j'appelerai le numéro de l'objet **INDICE**. Pour la mise en mémoire, j'utiliserai :

```
D$(1) = "CLEPTO"
D$(2) = "CHRONO"
D$(3) = "EPISTEMO"
D$(4) = "LOGO"
D$(5) = "NEO"
D$(6) = "PAN"
D$(7) = "PATHO"
```

En composant ces lignes, suivies d'un ordre de fin de message (Return ou Enter) je réalise l'opération d'**ENTRÉE DE DONNÉES.**

– Oui, dit Moneyback, et la LECTURE de ces données se fera, par exemple, en écrivant, en composant :

```
PRINT D$(4)
```

aussitôt la chaine LOGO apparait sur l'écran.

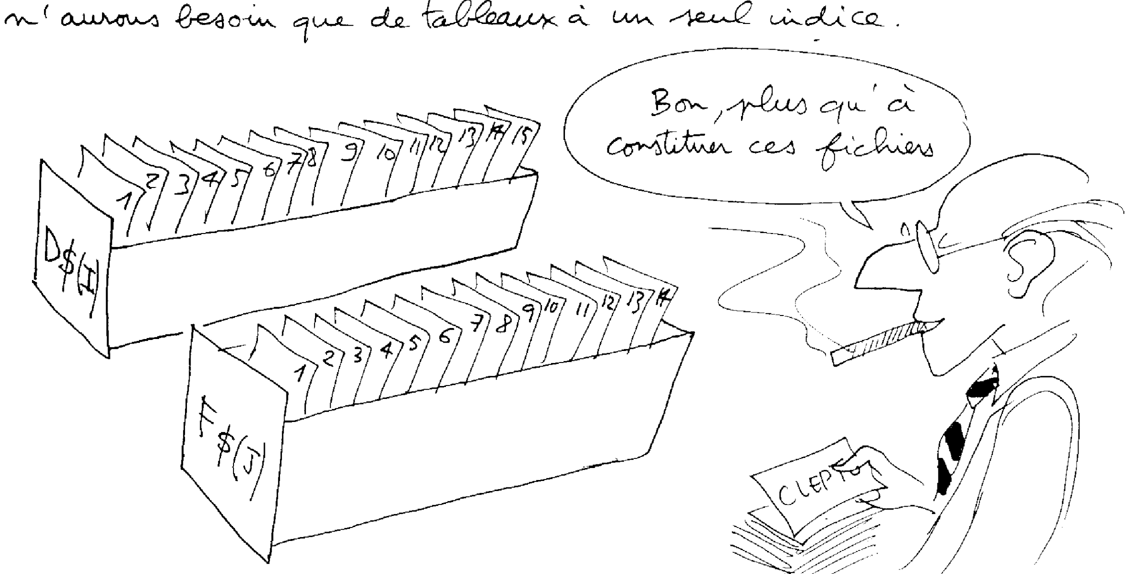
Sophie renchérit :

– Il est bien évident que je peux aussi écrire I = 4 ce qui indique de placer la valeur 4 dans le REGISTRE MEMOIRE I. Ensuite, avec PRINT D$(I) j'arriverai au même résultat.

Le visage de madame Piriardel se décrispa.

– Alors PRINT D$(I) veut simplement dire : "va chercher dans le tableau D$ le I^ème^ objet et imprime le sur l'écran.

– Exactement !... Et, dans un ordinateur, il peut exister des tableaux à plusieurs indices, qu'on écrirait alors par exemple T$(I,J). I étant le numéro de LIGNE et J le numéro de COLONNE. Mais, dans ce qui va suivre, nous n'aurons besoin que de tableaux à un seul indice.



PROBLÈME DE PLACE, DE TAILLE MÉMOIRE

– Es-tu sûr que la place disponible en mémoire est assez grande ? intervint Sophie.

– Ah oui, la taille de la mémoire d'un ordinateur est FINIE.

– On a déjà vu dans L'INFORMATIQUE PREMIER CONTACT qu'un ordinateur était avant tout une grande MÉMOIRE, partagée en **ROM** et en **RAM**. Dans la partie de mémoire appelée ROM, se trouvent des choses ineffaçables, données ou programmes. En général tout ce qui sert à l'ordinateur pour travailler (sa "culture générale" son "langage", ce qui lui permet d'INTERPRÈTER ce qu'on lui dit, les ordres reçus) sont dans des ROM.

Le reste, les RAM, représente la portion de mémoire à laquelle l'UTILISATEUR peut avoir accès, et où il peut stocker indifféremment PROGRAMME ou DONNÉES.

Une INSTRUCTION permet, dans le LANGAGE, de connaître à tout moment la place disponible. Selon les machines, ce sera l'instruction **MEM** ou **FRE(Ø)**.

– Il faudrait savoir !

– Les différences entre les ordinateurs BASIC s'atténuent de plus en plus.

rayon... antiquités



MEM

est l'instruction qui, sur le PICO ORDINATEUR PC 1251 permet de connaitre la place mémoire. Cette petite machine, qui pèse cent grammes, a une MÉMOIRE PERMANENTE. Ainsi, quand on l'allumme, sa mémoire peut très bien être encombrée par un programme, ou des données.

On effacera toute trace de programme en composant l'instruction **NEW**.

L'ordre **CLEAR**, lui, annulera toutes les DONNÉES éventuellement présentes dans des REGISTRES DE DONNÉES.

Ce que faisant, on aura complètement vidé la machine.

Pour les autres micro cités, il suffit de les éteindre et de les rallumer pour tout remettre à zéro.

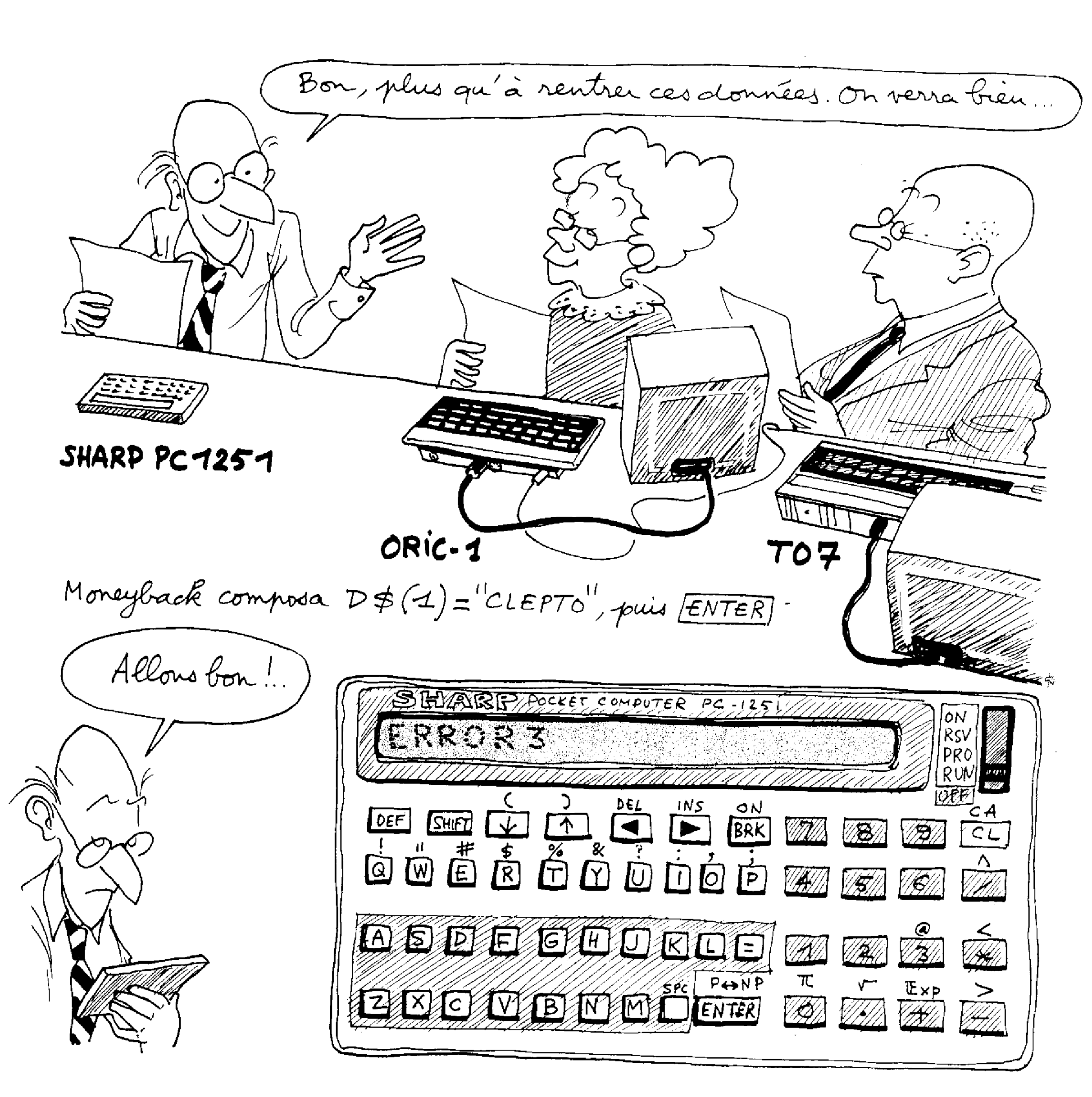
– Je compose maintenant MEM et [ENTER] (touche de fin de message) et je lis 3486.

– Maintenant allons vers l'ordinateur de table. Par exemple un **ORIC-1**. Cette fois ci tu composes **PRINT FRE(Ø)**

– Et j'obtiens 39421

– Tu vois que la place mémoire disponible dans un ORIC-1 est grosso modo dix fois plus grande que celle qu'offre un PC-1251.




Bon, plus qu'à rentrer ces données. On verra bien...
SHARP PC 1251
ORIC-1
TO7
Moneyback composa D$(1)="CLEPTO", puis ENTER.
Allons bon !..
SHARP POCKET COMPUTER PC-1251
ERROR 3



– C'est une question de RÉSERVATION DE PLACES MÉMOIRES commenta Cerise. Sur le PC1251 aucun stockage de fichier n'est admis si des places mémoires n'ont pas été au préalable réservées à cet effet.

– Et, comment fait-on pour faire cette réservation ?

– Rappelle toi, dans L'INFORMATIQUE, PREMIER CONTACT, nous avions déja évoqué cette instruction, page 67.

DIM

Cerise composa DIM D$(100)

– Et voilà, j'ai réservé cent emplacements mémoire pour créer le fichier D$(I) des débuts de mots. Ça n'est pas compliqué.

Moneyback fit **MEM** pour voir ce qui restait en mémoire, et obtint 1864.

– Sur les 3486 initiaux, j'en ai pris 1622. Voyons, si j'utilise mon PC 1251 comme fichier, combien de cases mémoire puis-je réserver ?

– Ça n'est pas difficile. Utilise l'instruction **CLEAR** pour annuler tes réservations de places en mémoire, et essaie DIM D$(200)

– MEM m'indique qu'il reste 264 places

– Continue ...

Moneyback fit CLEAR puis réserva un nombre croissant de places en mémoire.

– Ça marche jusqu'à DIM D$(216) inclu. Il reste alors 8 cases libres.

– En fait, dans le PC 1251, l'ordre DIM, quand il s'agit de fichiers de chaines de caractères, réserve des blocs de 16 caractères. Alors, compte : 16 × 216 + 8 = 3464. Tu retrouves, à très peu près le volume mémoire initialement disponible.

Sur cette machine, si tu veux, tu peux spécifier la longueur des chaines. Si, par exemple, tu veux retenir des ensemble de 11 caractères seulement, tu écriras DIM D$(216)*11

– Quel intérêt ?

– Sur une petite machine de poche comme celle là, tu gagnes de la place (sinon les registres D$(I) sont automatiquement dimensionnés à 16 caractères)

Moneyback fit CLEAR, puis DIM D$(216)*11, puis MEM et vérifia que Sophie disait vrai. Il restait 1093 cases vierges.

Dans les micro ordinateurs "de table" comme l'**ORIC-1** ou le **TO7** chaque lettre peut se transformer en fichier de dix éléments.

Comment cela ?

On peut donc "créer" sans précautions spéciales le fichier D$(Ø), D$(1), D$(2), D$(3), D$(4) D$(5), D$(6), D$(7), D$(8), D$(9)

Sur tous les ordinateurs BASIC, l'indice Ø est accepté. Mais, sans ordre de réservation préalable D$(13) sera refusé, par escemple.





— Mais non, monsieur Lemur, répondit Cerise, les ordinateurs possèdent des INSTRUCTIONS D'ENGENDREMENT DE HASARD. Et comme RANDOM en anglais veut dire hasard le nom de ces instructions sera **RANDOM** et **RND**.

Moneyback gratta son crâne dégarni.

— Il y a une chose qui m'étonne. Comment un ordinateur peut-il s'y prendre pour fabriquer du hasard, puisque c'est essentiellement une machine **DÉTERMINISTE** où toutes les choses obéissent à une logique rigoureuse, et à un programme fixé au départ. Il me semble y avoir une contradiction.

— Tu as parfaitement raison. L'ordinateur ne sait pas créer un **VRAI HASARD**, qui nécessite un PROCESSUS ALÉATOIRE.

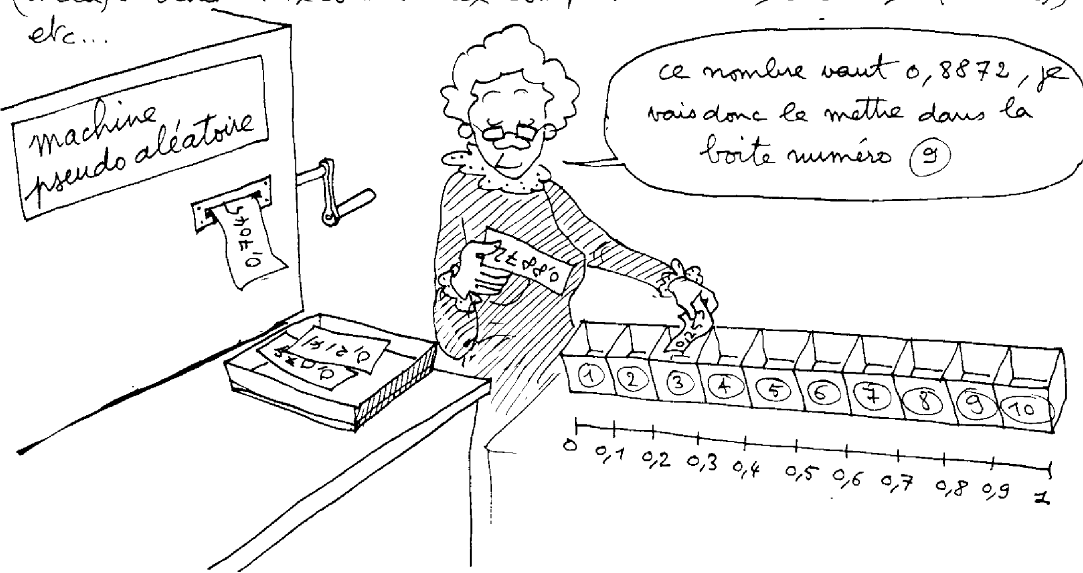
— C'est cela. Il n'y a pas de processus aléatoire dans le fonctionnement d'un ordinateur. Du moins à ce que j'en ai compris. Une machine obéissant à un PROGRAMME doit avoir un comportement PRÉVISIBLE

LE HASARD

— A l'intérieur des ordinateurs se trouve, à demeure (donc dans les **ROM**) un programme qui permet d'engendrer une suite de nombres **PSEUDO ALÉATOIRES.** Au lieu d'utiliser directement les instructions (RANDOM et RND) nous allons voir ce que recouvre ce programme.

— Oui, dit Moneyback, qu'est-ce qu'une suite de nombres pseudo-aléatoires et comment est-ce que cela se fabrique.

— Nous allons partir de nombres compris entre zéro et 1.
Je vais les classer dans des boîtes telles que :
→ dans la première boîte je mets les nombres compris entre 0 et 0,1 (inclu). Dans la seconde ceux compris entre 0,1 et 0,2 (inclu), etc...



–Si mon hasard est un "bon" hasard la distribution de ces nombres se fera de facon uniforme.

autrement dit, les boites vont contenir, grosso modo, le même nombre d'éléments

1 2 3 4 5 6 7 8 9 10

mais, comment s'y prend-t-on pour produire de tels nombres?

– C'est très simple. On va partir d'un premier nombre N, compris entre zéro et un, et qui jouera le rôle de **SEMENCE**.

A ce nombre **N**, j'ajoute **π** j'élève à la puissance cinq et je prends la partie fractionnaire du résultat

mais c'est une infâme cuisine !

- Exactement, monsieur lemur, je ne vous le fais pas dire. à l'aide de cette cuisine, j'obtiens un nouveau nombre compris entre zéro et un.
- Et je suppose que vous lui faites subir le même sort ?
- Précisément, et c'est comme cela que j'obtiens ma suite de nombres entre zéro et un.

Sophie écrivit le programme :

```
1Ø WAIT 1ØØ
2Ø PRINT "DONNEZ UN NOMBRE ENTRE"
3Ø INPUT "ZERO ET UN :"; N
4Ø IF N>1 THEN 1Ø
5Ø N = (N+π)^5 - INT((N+π)^5)
6Ø PRINT N
7Ø GOTO 5Ø
```

- A la ligne 5Ø on élève à la puissance cinq et on enlève la partie entière (instruction **INT**) ce qui donne la partie fractionnaire.

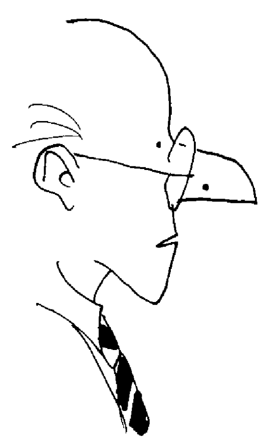
Moneyback, utilisant le PC1251, mit ce programme en mémoire. Pour ce faire il plaça le commutateur sur PRO. Il fit NEW pour effacer toute trace d'un programme précédent. **WAIT** est une instruction qui agit sur l'ordre PRINT, en créant un affichage non permanent (Wait veut dire attendre en anglais). La ligne 4Ø est un test avec SAUT CONDITIONNEL : si N est supérieur à 1 ALORS aller en 1Ø.
La ligne 7Ø crée un branchement inconditionnel vers la ligne 5Ø et met l'ordinateur en BOUCLE INFINIE.

Pour lancer le programme, je dois passer en mode RUN. Et je compose RUN [ENTER] au clavier

Sur mon ordinateur de table, je peux faire tout cela directement

– Sur les écrans apparaît DONNEZ UN NOMBRE ENTRE ZERO ET UN :

– Bon, j'introduis n'importe quoi, par exemple Ø.2561
les machines donnèrent immédiatement :



Ø.814636Ø12
Ø.185542155
Ø.71Ø587718
Ø.267972648
Ø.78153Ø925
Ø.3Ø8572164
Ø.876555839
Ø.4418244 3Ø
Ø.863ØØ5264
etc...

tu peux vérifier que cette distribution est uniforme, régulière

Effectivement, mais pourquoi pi ?
pourquoi la puissance 5 ?

– Pour partir d'une formule suffisamment compliquée. Au lieu de π j'aurais pu mettre un autre nombre, quelconque, comme 63.Ø7894123, et j'aurais pu élever à une puissance tout à fait quelconque, Répondit Sophie.

Mais, continua-t-elle, ceci n'est qu'une digression pour révéler les "dessous" de l'ordinateur. Nous avons des instructions qui font toutes seules ce travail. Si je veux par exemple créer un nombre pseudo aléatoire entre Ø et 56, je n'aurai qu'à écrire :

N = RND(1) * 56 sur l'ORIC-1

et N = RND 56 sur le SHARP PC 1251

l'ORIC.1 fournira, à chaque fois que je lancerai ce programme, une nouvelle volée de nombre.

Mais le PC1251 fournira toujours la même série.

– Comme si la "semence" de départ était toujours la même...

– Il existe une autre instruction RANDOM. Si elle précède dans le programme du PC 1251 l'instruction RND, alors les séries de nombres seront toutes différentes. On peut le vérifier en comparant les deux programmes :



– Revenons à nos moutons, dit Moneyback.
Nous avons donc des procédés simples qui nous permettent d'effectuer des tirages aléatoires des indices I et J des débuts et des fins de mots. A partir de là il doit être facile d'écrire le programme.

Il s'absorba un instant dans ses pensées et mit :

(*) BRK pour BREAK, qui veut dire arrêt, en anglais

```
100 RANDOM
102 REM "D = NB DEBUTS MOTS"
104 REM "F = NB FINS DE MOTS"
110 I = RND D
120 J = RND F
130 M$ = D$(I) + F$(J)
140 WAIT 100
150 PRINT M$
160 GOTO 110
```

tout cela a l'air d'un bon début ?..

MISE AU POINT DE PROGRAMME

Mais, est-ce que ça marche ?..

Sophie jeta un oeil sur ces quelques lignes.

– Souvent, en informatique, les programmes sont mis au point par une démarche ESSAI-ERREUR. Surtout quand on débute. On ne connait pas la machine qu'on manipule à fond, et on oublie des choses.

– Il manque déja une chose, dit Achille, ce sont les données. Inutile de faire tourner ce programme sans ses fichiers débuts de mots D$(I), et fins de mots F$(J). Je vais rajouter ce qu'il faut.

– avant de faire cela, tu peux déja voir la place mémoire occupée par ce programme.

Moneyback fit **MEM** sur le PC1251 et lut 3383

– La place prise est 3486 – 3383 = 103 "pas". C'est peu.

– Tu remarqueras que la place occupée par chaque ligne programme correspond au nombre de caractères utilisés. Tiens, rajoute par exemple la ligne :

```
1 REM "PROGRAMME LOGOTRONIQUE"
```

Une manoeuvre rapide indiqua que cette ligne avait mobilisé 28 nouveaux pas.

– le numéro de ligne représente à lui seul trois pas, même si le numérotage ne nécessite apparemment qu'un seul caractère.

– C'est comme si le numérotage correspondait à ØØ1

– les instructions, comme REM, GOTO, IF, etc... comptent pour un caractère.

– On retrouve le compte. Bon, ajoutons les données :

```
1 REM "PROGRAMME LOGOTRONIQUE"
2 DIM D$(12)
3 DIM F$(12)
1Ø D$(Ø)="CLEPTO"
11 D$(1)="CHRONO"
12 D$(2)="EPISTEMO"
13 D$(3)="LOGO"
14 D$(4)="NEO"
15 D$(5)="PAN"
16 D$(6)="PATHO"
17 D$(7)="HEMI"
18 D$(8)="CRYPTO"
19 D$(9)="ELASTO"
2Ø D$(1Ø)="EROTICO"
```

```
21 D$(11)="GNOSEO"
22 D$(12)="ORCHIDO"
3Ø F$(Ø)="GAME"
31 F$(1)="SE"
32 F$(2)="N"
33 F$(3)="EDRE"
34 F$(4)="DYNE"
35 F$(5)="CERQUE"
36 F$(6)="DROME"
37 F$(7)="PHOBE"
38 F$(8)="PATHE"
39 F$(9)="SCAPHE"
4Ø F$(1Ø)="DUC"
41 F$(11)="DENDRON"
42 F$(12)="ZOAIRE"
```

— Cette séquence va précéder celle de la page 25. A laquelle je rajoute

106 D = 12
108 F = 12

Intriqués, madame Pinardel et monsieur Lemur avaient également décidé de mettre ce programme dans leurs machines respectives. Cerise donna quelques précisions :

— Supprimez la ligne 100 et mettez :

110 I = RND (1) * D
120 J = RND (1) * F

— Maintenant, la minute de vérité, dit Moneyback.



— Ne t'énerve pas, oncle Achille. Ceci n'est dû qu'à une particularité de ta machine de poche. La ligne 130 fabrique, par **CONCATÉNATION** (catena en latin veut dire chaîne.) une chaîne de caractères M$ qui est censée être la réunion des deux chaînes D$(I) et F$(J). Mais, sur cette machine, sauf spécification, les chaînes de caractères sont limitées à sept caractères. D'où cette troncature, qui s'effectue automatiquement.
— Alors, que faire ?
— Première solution : supprimons cet intermédiaire M$, c'est à dire la ligne 130. Et écrivons :

```
150 PRINT D$(I);F$(J)
```

Ce qui veut dire "imprimer bout à bout".
— Effectivement, tout s'arrange...

Et Moneyback vit apparaître :

GNOSEODYNE
ORCHIDOSCAPHE
LOGODROME
PANPHOBE
CRYPTODUC
PATHOZOAIRE
CHRONODUC
EPISTEMOSE
ORCHIDOEDRE etc...

Cerise poursuivit :

– On aurait pu procéder autrement. En créant un fichier M$ n'ayant qu'un seul élément M$(∅). Les mots créés peuvent atteindre 22 caractères. Il me suffit de mettre 24 (la capacité d'affichage du PC1251) pour être tranquille

```
4 DIM M$(∅)*24
:    :    :
130 M$(∅)= D$(I) + F$(J)
140 WAIT 100
150 PRINT M$(∅)
160 GOTO 110
```

– Effectivement, ça marche !

Monsieur Lemur fit remarquer qu'avec deux fois treize racines (le ∅ compte), on pouvait engendrer 169 néologismes.

– Oui, dit Moneyback, mais la mémoire est loin d'être pleine. Voyez, il reste 2550 pas ! Pour savoir combien on peut loger de racines verbales dans cette machine, il suffit de faire le calcul. Je vais rajouter UNE racine, et voir quelle place cela me prend. Et il écrivit :

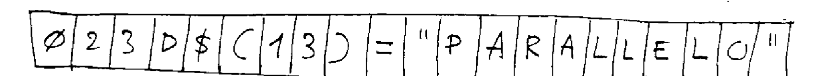
```
2 DIM D$(13)*11

23 D$(13) ="PARALLELO"

106 D =13
```

– Il reste 2518 pas. Tu en as consommé trente deux.

– Comment se fait-il ?



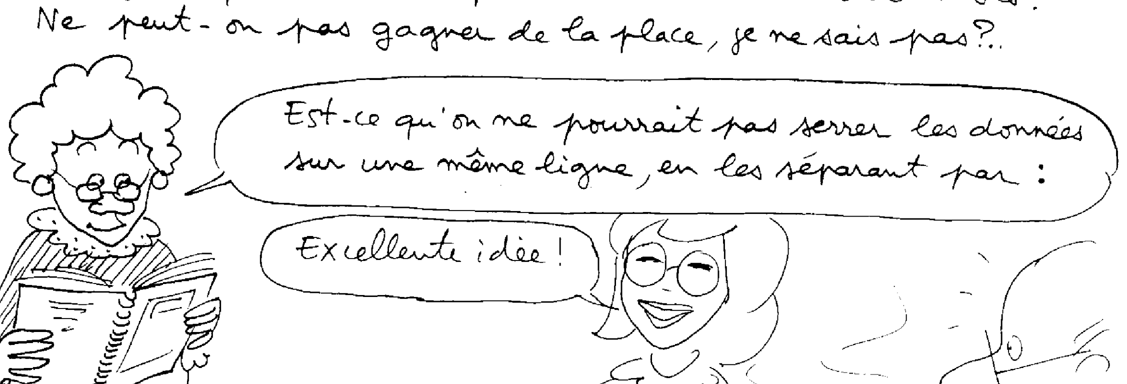
Je n'ai ajouté qu'une ligne programme de 21 caractères.

– Tu oublies les onze places mémoires réservées par ton ordre DIM.

– Ah oui... mais c'est un gâchis de place. De cette manière la DONNÉE est DEUX FOIS présente dans la mémoire.

– Eh oui .... et, dans cette façon de procéder, notre générateur de néologismes, sur PC 1251, ne pourra travailler que sur un peu plus d'une cinquantaine de racines verbales début et fin de mot.

– Ça fait quand même plus de 50×50=2500 mots. Ne peut-on pas gagner de la place, je ne sais pas?..



– Oui, oncle Achille, les lignes BASIC du PC 1251 peuvent contenir jusqu'à 80 caractères. Dans notre cas, cela permet de consigner cinq données par ligne :

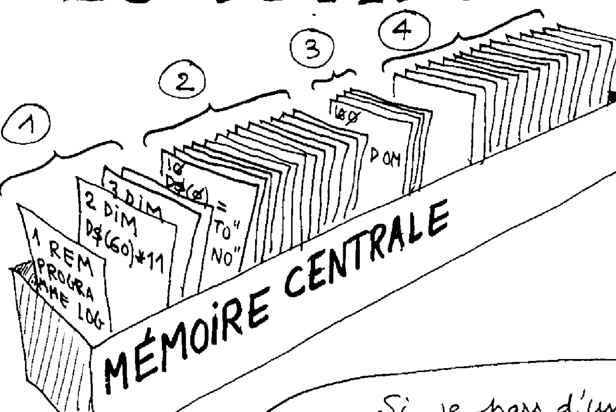
```
10 D$(0)="CLEPTO":D$(1)="CHRONO":D$(2)="EPISTEMO":
D$(3)="LOGO":D$(4)="NEO"
```

En opérant par différence, et en utilisant l'instruction MEM on peut voir qu'une telle ligne de données représente 78 pas. Ce à quoi il faut ajouter 5×11 = 55 pas réservés par le DIM soit 133 pas pour 5 données.
Le programme seul représente 189 pas et laisse 3297 pas libres.

– Ce qui équivaut à $3297 \times \frac{5}{133} = 123.94$, soit un peu plus de 120 données. Soixante débuts de mots et soixante fins de mots. ou $60 \times 60 = 3600$ mots potentiels. On gagne, mais pas tellement...



LES DATA



— Voyons, dit Sophie, comment est utilisée la place mémoire. Représentons la mémoire CENTRALE de notre ordinateur (par opposition à sa MÉMOIRE EXTERNE dont le support est une bande magnétique ou une diskette).

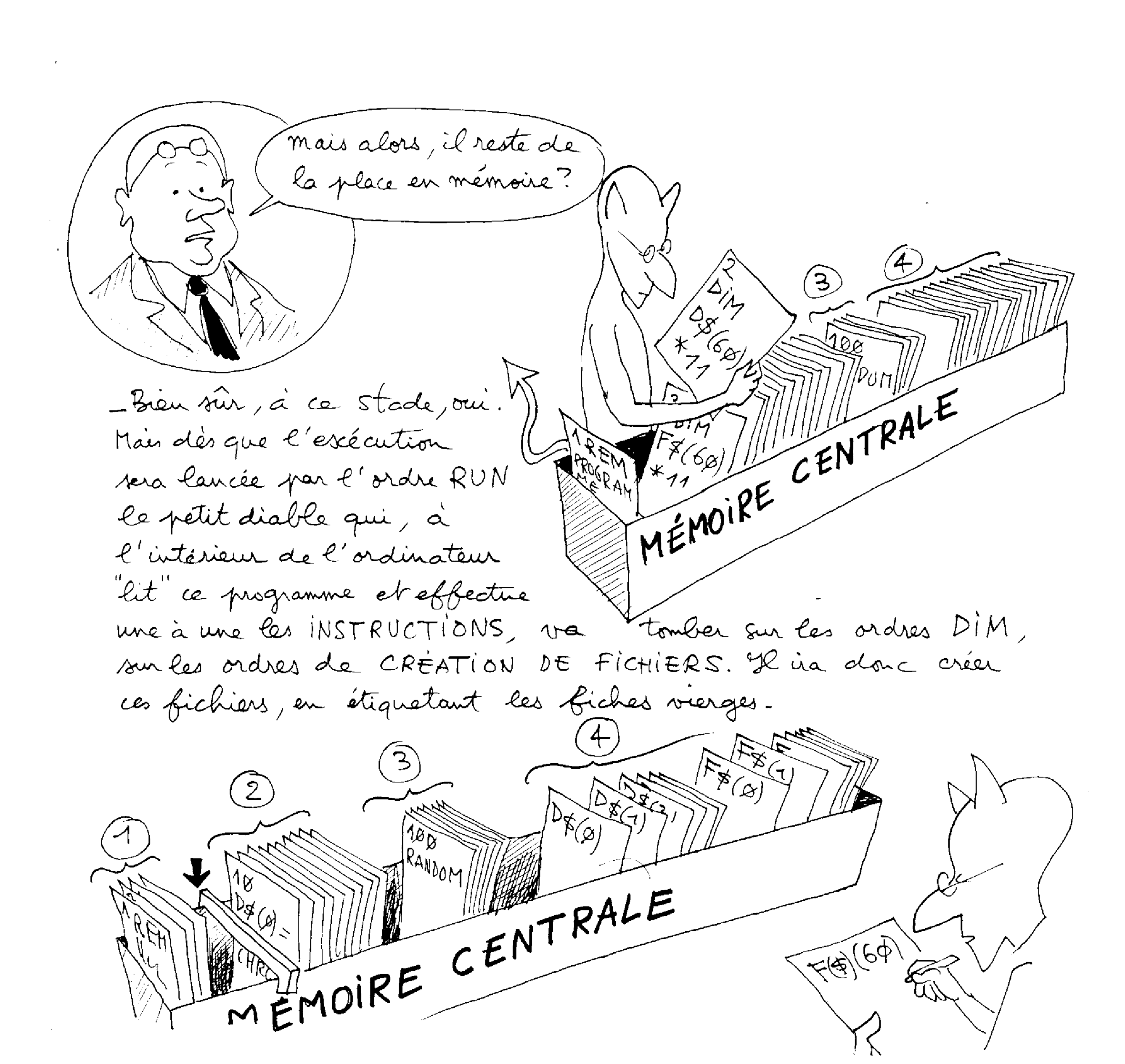
Si je pars d'une mémoire totalement vidée (NEW, puis CLEAR) et que je tape mon programme, la place mémoire sera affectée comme ceci :

① Titre du programme (un ordre REM) et instructions DIM (création des fichiers)

② Présence des données dans des lignes-programme.

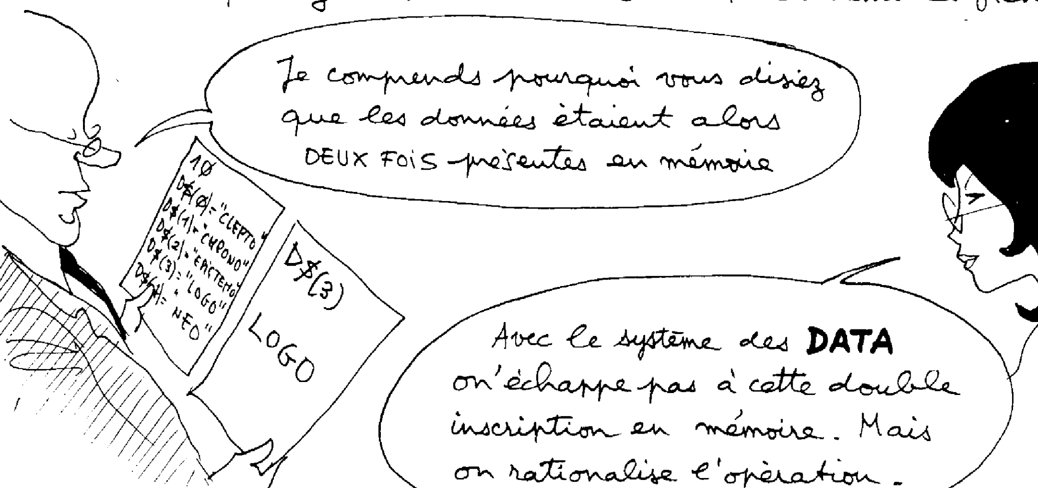
③ La boucle de travail de notre programme

④ Emplacement vierge.


Mais alors, il reste de la place en mémoire?
2 DIM D$(6Ø) *11
③
④
100 DUM
1 REM PROGRAM
DIM F$(6Ø) *11
MÉMOIRE CENTRALE
_Bien sûr, à ce stade, oui. Mais dès que l'exécution sera lancée par l'ordre RUN le petit diable qui, à l'intérieur de l'ordinateur "lit" ce programme et effectue une à une les INSTRUCTIONS, va tomber sur les ordres DIM, sur les ordres de CRÉATION DE FICHIERS. Il va donc créer ces fichiers, en étiquetant les fiches vierges.
①
②
③
④
10 D$(Ø)=
100 RANDOM
D$(Ø)
D$(1)
F$(Ø)
F$(1)
MÉMOIRE CENTRALE
F($)(6Ø)


D$(24)
J'ai bien compris : après exécution de ces ordres DIM, les registres D$(I) et F$(J) EXISTENT, mais ils sont comme des tiroirs vides
— C'est la séquence suivante qui va remplir ces tiroirs - la ligne 1Ø :
1Ø D$(Ø)="CLEPTO":D$(1)="CHRONO":D$(2)="EPISTEMO":
D$(3)="LOGO":D$(4)="NEO"
signifiant :
mettre la chaîne CLEPTO dans le registre D$(Ø)
mettre la chaîne CHRONO dans le registre D$(1)
mettre la chaîne EPISTEMO dans le registre D$(2)
etc...
1
2
3
4
1 REM PROGRAMME
3 DIM
2Ø
D$(5)="PAN
D$(6)="PATHO
HEM
1ØØ RANDOM
D$(1)
MÉM
NTRALE
1Ø
D$(Ø)="CLEPTO"
D$(1)="CHRONO"
D$(2)="EPISTEMO"
D$(3)="LOGO"
D$(4)="NEO"
D$(Ø)
CLEPTO

– Lorsque cette portion ② du programme a été exécutée, toutes les données qui s'y trouvaient ont été RECOPIÉES dans les fichiers.



– Bon, dit Sophie, reprenons tout à zéro.
Elle fit NEW (pour effacer le programme) et CLEAR (pour effacer les données).

Je garde les instructions de dimensionnement :

```
2 DIM D$(6Ø)*11
3 DIM F$(6Ø)*11
4 DIM M$(Ø)*24
```

qui vont réserver les places mémoire pour les fichiers.

– Je ne sais pas a priori combien je vais pouvoir mettre de données. Pour commencer, disons 60 débuts de mots et 60 fins de mots. On retouchera après, si besoin est. Et j'écris

200 DATA CLEPTO, CHRONO, EPISTEMO, LOGO, NEO, PAN, PATHO, HEMI, CRYPTO

Cette introduction de ligne programme ne posa aucun problème sur l'ORIC-1 ou le TO7. Mais, sur l'écran à cristaux liquides du SHARP PC 1251, Moneyback vit apparaître :

200 DATA CLEP TO, CHR ON O, E PI STEMO, LOG O, NEO, PAN, PATHO, HEMI, CRYP TO



– Cela n'est rien, oncle Achille. C'est un défaut du PC1251.
Il ne peut pas s'empêcher de reconnaître des **MOTS.CLEFS** du **BASIC** dans les **DATA**. Ainsi, dans CLEPTO se trouve **TO**, lié aux instructions de boucle **FOR ..... TO ....**
Dans CHRONO on trouve **ON**
Dans EPISTEMO ..... les lettres composant **PI** !
Dans LOGO la suite **LOG** (Logarithme).
– Eh bien, cela fait une belle salade !
– Pour t'en sortir, utilise des guillemets (♥)
Et Sophie écrivit :

```
200 DATA "CLEPTO","CHRONO","EPISTEMO","LOGO","NEO",
"PAN","PATHO","HEMI","CRYPTO"
210 DATA "EROTICO","GNOSEO","ORCHIDO","PARALLELO",
"SEMIO","ERGO","GERONTO","NECRO"
220 DATA "MESO","PODO","PORNO","PROTO","SCATO","DOXO",
"PLOUTO","PHONO","INFLATO"
230 DATA "PYRO","RETRO","PSYCHO","LITHO","MACRO",
"CHRYSO","PHILO","MICRO","TECHNO"
240 DATA "BIO","SPELEO","TOMO","TOPO","TELEO","RHINO","PHYTO",
"PHOTO","HETERO"
250 DATA "TERATO","THERMO","BUTYRO","PHRENO"
260 DATA "ELASTO","THEO","VIDEO","PAPYRO","CEPHALO",
"NECRO","MYCO","*"
```

(♥) Inutiles pour les ordinateurs "de table" ORIC-1, TO7, etc...

```
3ØØ DATA "GAME","SE","N","EDRE","DYNE","CERQUE","DROME",
"PHOBE","PATHE","SCAPHE"
31Ø DATA "TOPE","TRON","ME","METRE","MORPHE","NAUTE",
"DRAME","IDE","OYNE","COSME"
32Ø DATA "LOGUE","LOGIE","SOME","SPHERE","STASE","TAPHE",
"TROPE","POLITIQUE","MANE"
33Ø DATA "GRADE","GRAMME","GENE","ELASTIQUE","CINESE",
"CRATE","DYNAMIQUE","CYTE"
34Ø DATA "PHRENE","LYSE","MACHIE","PHAGE","CEPHALE",
"THEISME","PHYSIQUE"
35Ø DATA "THERAPEUTHE","PSYCHOLOGIE","UNIVERS","PROPUL-
SION","EROTISME","GRAPHE"
36Ø DATA "PHILOSOPHIE","CENTRIQUE","COSMOS","CYCLE",
"ELECTRICITÉ","CHRONE","DIDACTE"
37Ø DATA "DUC","DENDRON","ZOAIRE","SAURE","RRHÉE",
"MANE","MANCIE","SEXUEL"
38Ø DATA "THEQUE","SOCIOLOGIE","LATRE","PITHEQUE","*"
```

Très bien, mais comment est-ce que ces données vont venir prendre place dans les registres des fichiers D$(I) et F$(J) ?

Il faut écrire une séquence de lecture des DATA

READ

(Read veut dire lire en anglais).

```
30  I = -1                    <== initialisation compteur I
40  I = I+1                   <== Compteur de préfixes: incrémentation
50  READ D$(I)                <== lecture de data
60  IF D$(I)="*"THEN 80       <== Fin de lecture de data
70  GOTO 40                   <== Retour en boucle
80  D = I-1                   <== Sortie de boucle
90  J = -1                    <== initialisation compteur J
100 J = J+1                   <== Compteur de suffixes, incrémentation
110 READ F$(J)                <== lecture des data suivants
120 IF F$(J)="*"THEN 140      <== Fin de lecture des data
130 GOTO 100                  <== Retour en boucle
140 F = J-1                   <== Sortie de boucle
```

- Voici la séquence de lecture de ces data - Deux registres, I et J, sont utilisés comme compteurs. La ligne 30 donne à I la valeur initiale - 1.

A la ligne 40 on réalise une **INCRÉMENTATION**, où le registre I est accru d'une unité (**INCRÉMENT = 1**)

Quand l'instruction **READ** est rencontrée, la première donnée figurant dans la suite des DATA (lignes 200 à 370) vient se placer dans le registre suivant l'instruction READ. En l'occurence ici D$(I).

Dans cette suite de DATA on a mis deux caractères-repères *

(on a pris un astérisque, mais tout autre caractère aurait pu convenir). la ligne 6∅ :

```
6∅ IF D$(I)="*" THEN 8∅
```

permet de sortir de la boucle, puisqu'elle signifie :

"Si D$(I) = "astérisque" ALORS aller en 8∅

Ce faisant, on a crée un "parasite" dans le fichier D$(I) et il est nécessaire d'adapter la valeur borne D à I-1.

(Précisons, comme l'indice ∅ compte, qu'il y aura alors D+1 débuts de mots dans le fichier D$(I)).

Opération analogue dans les lignes 1∅∅ à 14∅ pour alimenter le fichier F$(J) des fins de mots, qui contiendra F+1 éléments.



– Bon, on a introduit ce programme dans l'ordinateur préalablement vidé (NEW et CLEAR). MEM donne 2188.

Faisons RUN
– Ah, message d'erreur à la ligne 11∅

```
11∅ READ F$(J)
```

Tout a l'air correct, pourtant ?
– Voyons ce que vaut J
– J vaut 61
– Rappele-toi. Nous n'avions réservé que 6∅ places mémoire pour chaque fichier. Comptons nos data.
– Il y a 55 débuts de mots et 69 fins de mots.
– Plus les astérisques de marquage. Il faut réserver 56 places pour le fichier D$(I) et 70 pour le fichier F$(J), en mettant :

```
2  DIM D$(55)*11
3  DIM F$(69)*11
```

– Mais, pourquoi pas DIM D$(56) et DIM F$(70) ?
– Tu peux le faire, mais comme l'indice ∅ compte, DIM D$(55) et DIM F$(69) réservent en fait 56 et 70 places mémoires.
– Ah, c'est vrai ...

Tous effectuèrent la correction.
– Et maintenant, plus de message d'erreur. Et le SHARP indique qu'il reste 760 places libres

– Maintenant rajoutons la séquence de création de néologismes :

SHARP PC 1251

```
150 RANDOM
160 I = RND D : J = RND F
170 M$(0) = D$(I) + F$(J)
180 WAIT 100 : PRINT M$(0)
190 GOTO 160
```

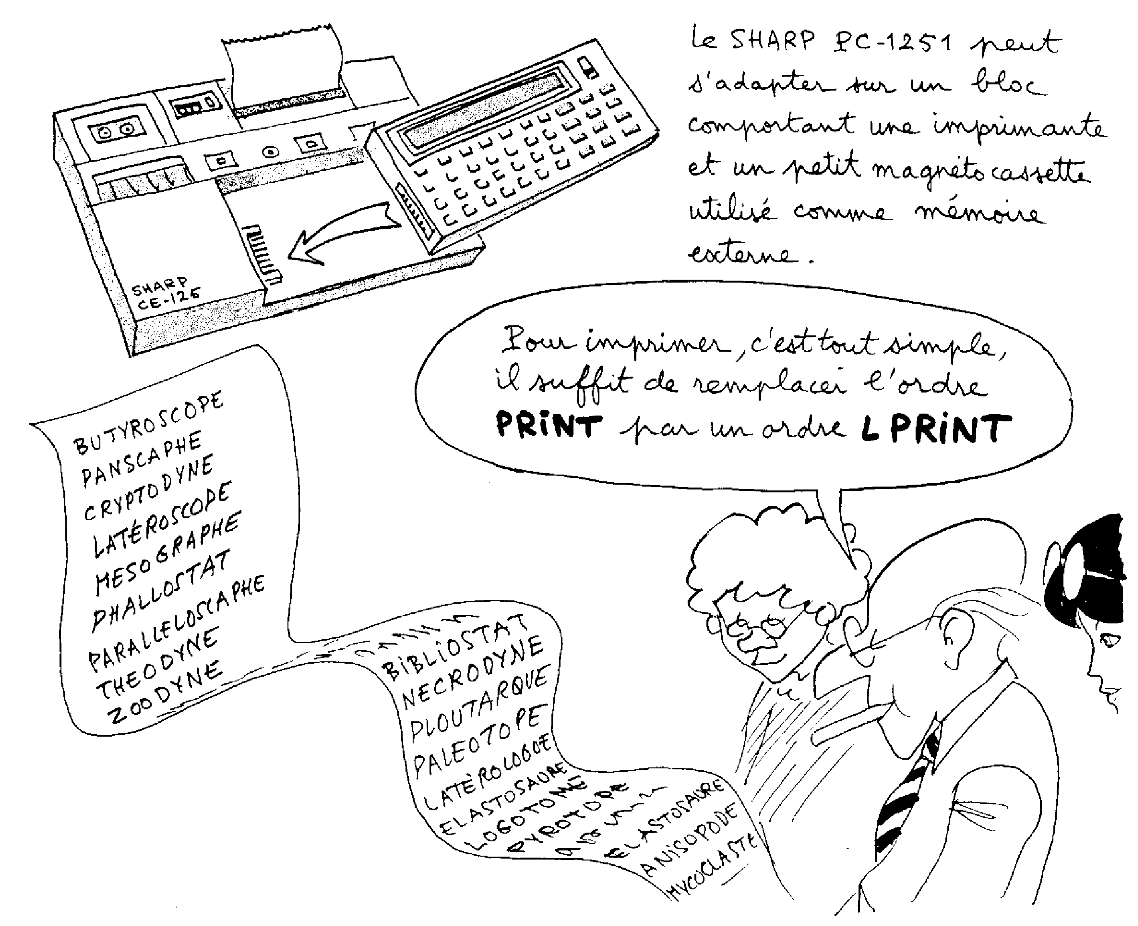
ORIC-1 TO7 etc...

```
150 I = RND(1) * D
160 J = RND(1) * F
170 M$ = D$(I) + F$(J)
180 WAIT 100 : PRINT M$
190 GOTO 160
```

– chaque donnée supplémentaire, dans notre problème particulier, nous coûte en moyenne une vingtaine de pas. Nous avons réussi avec les data à entrer 69 + 55 = 124 racines verbales. Comme il reste 760 pas, on devrait pouvoir monter à 160 données, 160 racines. Soit 80 débuts de mots et 80 fins de mots
– autrement dit 6400 mots potentiels.

➔ Complètez ce programme en rajoutant des données. N'oubliez pas de décaler les astérisques et d'adapter les ordres DIM en conséquence...

LE LOGOTRON


Le SHARP PC-1251 peut s'adapter sur un bloc comportant une imprimante et un petit magnéto cassette utilisé comme mémoire externe.
SHARP CE-125
Pour imprimer, c'est tout simple, il suffit de remplacer l'ordre **PRINT** par un ordre **LPRINT**
BUTYROSCOPE
PANSCAPHE
CRYPTODYNE
LATÉROSCOPE
MESOGRAPHE
PHALLOSTAT
PARALLELOSCAPHE
THEODYNE
ZOODYNE
BIBLIOSTAT
NECRODYNE
PLOUTARQUE
PALEOTOPE
ELASTOSAURE
ELASTOSAURE
ANISOPODE
MYCOCLASTE

VOICI UN LOT DE RACINES VERBALES OÙ VOUS POURREZ PUISER LES ÉLÉMENTS DE VOTRE LOGOTRON :

DÉBUTS DE MOTS :

1 ACRO : EXTREMITE
2 ADENO : GLANDE
3 AERO : AIR
4 AGRO : CHAMP
5 ALGO : DOULEUR
6 ALLO : AUTRE
7 ANDRO : HOMME
8 ARITHMO : NOMBRE
9 ARRHENO : MALE
10 ARSENO : VIRIL
11 ANTE : AVANT
12 ANTI : CONTRE
13 ANTHO : FLEUR
14 ARTERIO : ARTERE
15 ARTHRO : ARTICULATION
16 ASTERO : ETOILE
17 ASTRO : ASTRE
18 ASTHENO : FATIGUE
19 ANTHROPO : HOMME
20 APO : A PARTIR DE
21 APHRO : MOUSSE
22 ARCHEO : ANCIEN
23 ARGO : BRILLANT
24 BACTERIO : BACTERIE
25 BALANO : GLANDE
26 BARO : RRESSION
27 BARY : LOURD
28 BATHY : PRESSION
29 BIBLIO : LIVRE
30 BIDULO : MACHIN
31 BIO : VIE
32 BLENO : MUCUS
33 BLASTO : GERME
34 BOLO : JETER
35 BRACHY : COURT
36 BRADY : LENT
37 BRONCHO : BRONCHE
38 BRONTO : TONNERRE
39 BUTYRO : BEURRE
40 CACO : MAUVAIS
41 CARDIO : CŒUR
42 CARYO : NOIX
43 CATA : SUR
44 CENTRO : CENTRE
45 CERCO : QUEUE
46 CINEMATO : MOUVEMENT
47 CLEPTO : DÉROBER
48 CLIMATO : CLIMAT
49 CLINO : INCLINÉ
50 CENO : VIDE
51 CEPHALO : TÊTE
52 CERATO : CORNE
53 CHEIRO : MAIN
54 CHELI : PINCE
55 CHETO : CRINIÈRE
56 CHIMIO : CHIMIE
57 CHLORO : JAUNE
58 CHRYSO : OR
59 CHROMO : COULEUR
60 CHRONO : TEMPS
61 CO : AVEC
62 CŒLO : CREUX
63 COPRO : EXCRÉMENT
64 CONTRA : CONTRE
65 COSMO : COSMIQUE
66 CRANIO : CRANE
67 CRISTALLO : VERRE
68 CRYO : FROID
69 CRYPTO : CACHÉ
70 CYANO : BLEU
71 CYCLO : CERCLE
72 CYNO : CHIEN
73 CYPHO : CONVEXE
74 CYTO : CELLULE
75 DACTYLO : DOIGT
76 DECI : DIXIÈME
77 DERMATO : PEAU
78 DI : DOUBLE
79 DIA : A TRAVERS
80 DIPLO : DOUBLE
81 DINO : TERRIBLE
82 DECA : DIX
83 DEMO : PEUPLE
84 DEMONO : DIABLE
85 DENDRO : ARBRE
86 DERMO : PEAU
87 DEUTERO : SECOND

88 DEXTRO : DOIGT
89 DODECA : DOUZE
90 DOXO : OPINION
91 DRAMA : THÉATRE
92 DROMO : COURIR
93 DYNAMO : ÉNERGIE
94 DYS : DIFFICULTÉ
95 ECTO : À L'EXTÉRIEUR
96 ELASTO : ÉLASTIQUE
97 ELECTRO : ÉLECTRICITÉ
98 EMBRYO : EMBRYON
99 ENCEPHALO : CERVEAU
100 ENDO : À L'INTÉRIEUR
101 ENTERO : ENTRAILLES
102 ENTOMO : INSECTE
103 EO : AURORE
104 EPI : SUR
105 EPISTEMO : SCIENCE
106 ERGO : TRAVAIL
107 ESTHETICO : ESTHETIQUE
108 EROTICO : LIÉ A L'AMOUR
109 ETHYMO : VRAI
110 EXO : AU DEHORS
111 EXTRA : AU DEHORS
112 FANTASMO : ILLUSION
113 FIBRO : FIBRE
114 GALACTO : LAIT
115 GAMO : MARIAGE
116 GASTRO : VENTRE
117 GEO : TERRE
118 GERONTO : VIELLARD
119 GENO : RACE
120 GLOSSO : LANGUE
121 GLYCO : SUCRE
122 GNOSEO : CONNAISSANCE
123 GONIO : ANGLE
124 GONO : SEMENCE
125 GRAMMO : MESSAGE
126 GRAPHO : ÉCRITURE
127 GRAVITO : GRAVITÉ
128 GYMNO : NU
129 GYNO : FEMME
130 GYRO : CERCLE
131 HALO : SEL
132 HECTO : CENT
133 HELIO : SOLEIL
134 HEMATO : SANG
135 HEMI : À MOITIÉ
136 HEPATO : FOIE
137 HEPTA : SEPT
138 HETERO : AUTRE
139 HEXA : SIX
140 HIERO : SACRÉ
141 HIPPO : CHEVAL
142 HISTO : TISSUS
143 HODO : CHEMIN
144 HOLO : ENTIER
145 HOMEO : SEMBLABLE
146 HOMO : SEMBLABLE
147 HYDRO : EAU
148 HYGRO : HUMIDITÉ
149 HYLO : MATIÈRE
150 HYMENO : MEMBRANE
151 HYPER : AU DELÀ
152 HYPHO : TISSU
153 HYPO : INSUFFISANCE
154 HYSTERO : UTERUS
155 ICHTIO : POISSON
156 ICONO : IMAGE
157 IDEO : IDÉE
158 IDIO : PROPRE
159 INFLATO : GONFLER
160 INFRA : EN DESSOUS
161 ISO : EGAL
162 INTRA : EN DEDANS
163 JUXTA : À CÔTÉ
164 KILO : MILLE
165 LATERO : CÔTÉ
166 LARYNGO : GORGE
167 LEPTO : MINCE
168 LEVO : GAUCHE
169 LIPO : GRAISSE
170 LITHO : PIERRE
171 LOGO : DISCOURS
172 LOXO : OBLIQUE
173 MACRO : GRAND
174 MAGNETO : MAGNÉTISME
175 MASTO : MAMMELLE
176 MEGALO : GRAND
177 MELO : CHANT
178 MERO : PARTIE
179 MESÓ : MILIEU
180 METEO : ELEVÉ
181 META : CHANGEMENT
182 METRO : MESURE
183 MICRO : PETIT
184 MISO : HAÏR
185 MNEMO : MÉMOIRE
186 MORPHO : FORME
187 MYCO : CHAMPIGNON
188 MYELLO : MOELLE
189 MYO : MUSCLE
190 MYTHO : MYTHE
191 NECRO : MORT
192 NEO : NOUVEAU
193 NEURO : NERFS
194 NEMATO : FIL
195 NEVRO : NERF

196 NOO : ESPRIT
197 NOMO : LOI
198 NOSO : MALADIE
199 NYCTO : NUIT
200 NUCLEO : NOYAU
201 OCTO : HUIT
202 ODO : ROUTE
203 OLEO : HUILE
204 OLIGO : PEU NOMBREUX
205 OMNI : TOUT
206 ONTO : L'ÊTRE
207 OO : ŒUF
208 OPHTALMO : ŒIL
209 ORCHIDO : COUILLES
210 ORGANO : ORGANE
211 ORNITHO : OISEAU
212 ORO : MONTAGNE
213 ORTHO : DROIT
214 OSCILLO : OSCILLER
215 OSMO : PULSION
216 OTO : OREILLE
217 OVO : ŒUF
218 PACHY : ÉPAIS
219 PALEO : ANCIEN
220 PAN : TOUT
221 PARA : VOISIN DE
222 PANTO : TOUT
223 PAPYRO : PAPIER
224 PARELLELO : PARALLELE
225 PATA : PATA
226 PATHO : SOUFFRANCE
227 PEDO : PIED
228 PEDO : SOL
229 PENTA : CINQ
230 PERI : AUTOUR DE
231 PETRO : PIERRE
232 PHAGO : MANGER
233 PHALLO : ZIZI
234 PHANERO : VISIBLE
235 PHARMACO : MEDICAMENT
236 PHARINGO : GOSIER
237 PHENO : APPARAITRE
238 PHILO : AIMER
239 PHOBO : PEUR DE
240 PHONO : SON
241 PHOTO : LUMIÈRE
242 PHRENO : ESPRIT
243 PHYLO : RACE
244 PHYSIO : NATURE
245 PHYTO : PLANTE
246 PINACO : TABLEAU
247 PITHECO : SINGE
248 PLASMO : FACONNER
249 PLASMO : MODELAGE
250 PLECTO : SOUDER
251 PLEO : PLUS
252 PLEISTO : BEAUCOUP
253 PLESIO : PROCHE
254 PLEURO : A COTÉ
255 PLOUTO : RICHE
256 PLURI : PLUSIEURS
257 PNEUMO : POUMON
258 PODO : PIED
259 POLY : NOMBREUX
260 PORNO : COCHON
261 POST : APRÈS
262 PRIMO : PREMIER
263 PRO : DEVANT
264 PROTO : PREMIER
265 PSEUDO : FAUX
266 PSYCHO : AME
267 PTERO : AILE
268 PYO : PUS
269 PYRO : FEU
270 QUASI : PRESQUE
271 RADIO : RADIO, RAYON
272 RETRO : EN ARRIÈRE
273 RHEO : COULER
274 RHINO : NEZ
275 RHIZO : RACINE
276 RHOMBO : LOSANGE
277 SCATO : EXCRÉMENT
278 SCLERO : DUR
279 SELENO : LUNE
280 SEMIO : SENS
281 SEMA : SIGNE
282 SEMI : À DEMI
283 SEXO : SEXE
284 SCHIZO : FENDRE
285 SIDERO : FER
286 SIMILI : SEMBLABLE
287 SOLENO : TUYAU
288 SOMATO : CORPS
289 SONO : SON
290 SPÉLÉO : CAVERNE
291 SPHENO : COIN
292 SPHERO : SPHERE
293 SPLENO : RATE
294 SPONDYLO : VERTEBRE
295 STAPHILO : GRAIN DE RAISIN
296 STEGO : TOIT
297 STENO : ÉTROIT
298 STEREO : SOLIDE
299 STERNO : POITRINE
300 STOMATO : BOUCHE
301 STROBO : TOURBILLON
302 STYLO : COLONNE
303 SUB : SOUS

304 SULFANO : SOUFRE
305 SYL : AVEC
306 SYM : ENSEMBLE
307 SYN : AVEC
308 TACHY : VITESSE
309 TAUTO : LE MÊME
310 TAXO : ORDRE
311 TECHNO : ART
312 TELEO : LOIN
313 TERATO : MONSTRE
314 TETRA : QUATRE
315 THALASSO : MER
316 THEO : DIEU
317 THERMO : CHALEUR
318 TOMO : COUPER
319 TONI : TON
320 TOPO : Lieu
321 TRIBO : FROTTER
322 TROPO : TENDANCE
323 TYPO : CARACTÈRE
324 ULTRA : AU DELA'
325 URANO : CIEL
326 URO : URINE
327 VICE – ; VICE –
328 VICE ; A LA PLACE DE
329 VIDEO : VOIR
330 VISIO : VISION
331 XENO : ETRANGER
332 XYLO : BOIS
333 ZOO : ANIMAL

FINS DE MOTS

1 ACRE : EXTRÉMITE
2 AGOGIE : QUI CONDUIT
3 AGOGUE : QUI CONDUIT
4 ALGIE : DOULEUR
5 ARQUE : COMMANDER
6 ANTHE : FLEUR
7 ASTERE : ÉTOILE
8 ASTRE : ASTRE
9 ASTHENIE : FATIGUE
10 ANTHROPE : HOMME
11 ARCHIE : COMMANDER
12 ARGE : BRILLANT
13 BACTERIE : BACTERIE
14 BARE : PRESSION
15 BATHE : PROFOND
16 BIE : VIE
17 BLASTE : GERME
18 BOLE : JETER
19 CARDE : CŒUR
20 CARPE : PAUME
21 CENTRIQUE : CENTRE
22 CERE : CORNE
23 CERQUE : QUEUE
24 CINES : MOUVEMENT
25 CINETIQUE : “ ”
26 CLINE : PENTE
27 CELE : TUMEUR
28 CENE : VIDE
29 CEPHALE : TÊTE
30 CHIMIE : CHIMIE
31 CHRYSE : OR
32 CHROME : COULEUR
33 CHTONE : TEMPS
34 CHTONE : TERRE
35 CLASTE : BRISER
36 COQUE : GRAINE
37 COSME : COSMOS
38 COSMIQUE
39 CRATE : FORCE
40 CULTEUR : CULTURE
41 CULTURE
42 CYCLE : CERCLE
43 CYTE : CELLULE
44 DACTYLE : DOIGT
45 DERME : PEAU
46 DIDACTE : INSTRUIRE
47 DES : RÉUNIR
48 DOX : OPINION
49 DRAME : THEATRE
50 DROME : COURIR
51 DROMIE : “ ”
52 DONTE : DENT
53 DUC : CONDUIRE
54 DYNE : ENERGIE
55 ECTOMIE : ABLATION
56 ÈDRE : BASE
57 ELASTIQUE : ELASTIQUE
58 ELECTRIQUE : id
59 EMIE : SANG
60 FANTASME : ILLUSION
61 FUGE : FUIR
62 FIBRE : id
63 : GAME : mariage
64 GASTRE : VENTRE
65 GASTERE : id
66 GÉE : TERRE

67 GENESE : FORMATION
68 GENE : " "
69 GLOTTE : LANGUE
70 GLYPHE : GRAVURE
71 GNATHE : MÂCHOIRE
72 GNOSE : CONNAISSANCE
73 GONE : ANGLE
74 GRADE : MARCHER
75 GRAMME : MESSAGE
76 GRAPHE : ÉCRIRE
77 GRAVITATION
78 GRAVITATIONNEL
79 GRAVITÉ
80 GYNE : FEMME
81 GYRE : CERCLE
82 HELIE : SOLEIL
83 HEPATIQUE : FOIE
84 HYDRE : EAU
85 IATRE : MÉDECIN
86 IDE : FORME
87 INFLATION : GONFLEMENT
88 INFORMATIQUE : ID
89 LATÈRE : CÔTÉ
90 LATRE : ADORATEUR
91 LITHE : PIERRE
92 LOGIE : DISCOURS SUR
93 LOGUE : " "
94 LYSE : COUPER
95 MACHIE : COMBAT
96 MAGNETISME
97 MAGNETIQUE
98 MANE : FOU DE
99 MANCIE : PRÉDICTION
100 DE L'AVENIR PAR
101 MANIE : FOLIE DE
102 MATHEMATIQUES
103 MATIQUE : MOUVEMENT
104 MATE : QUI SE MEUT
105 ME
106 MEGALIE : GRAND
107 MERE : PARTIE DE
108 MÈTRE : MESURE
109 MNESIE : MÉMOIRE
110 MORPHE : FORME
111 MORPHISME
112 MYCETE : CHAMPIGNON
113 MYCOSE : ID
114 MYTHE : MYTHE
115 NAUTE : QUI NAVIGUE
116 NÉSIE : ÎLE
117 NÉVROSE : NERF
118 NOME : LOI
119 NOMIE : ID
120 NYMO : NOM
121 NUCLÉIQUE : NOYAU
122 ORCHIDE : COUILLE
123 PATHIE : SOUFFRANCE
124 PEDE : PIED
125 PETE : QUI VA VERS
126 PHAGE : MANGER
127 PHANIE : APPARITION
128 PHASE : " "
129 PHILE : QUI AIME
130 PHOBE : QUI DÉTESTE
131 PHONE : SON
132 PHORE : QUI PORTE
133 PHOTE : LUMIÈRE
134 PHRENE : ESPRIT
135 PHYLAXIE : GARDER
136 PHYLE : RAMEAU
137 PHYLLE : FEUILLE
138 PHYSE : GLANDE
139 PHYSIQUE : ID
140 PHYTE : PLANTE
141 PITHÈQUE : SINGE
142 PLASME : FAÇONNER
143 PLASTE : MODELAGE
144 PLASTIQUE " "
145 PNÉE : RESPIRATION
146 POIETIQUE : ACTION
147 DE FABRIQUER
148 PODE : PIED
149 POLE : VILLE
150 POLITIQUE
151 PROPULSION
152 PSYCHOLOGIE
153 PTÈRE : AILE
154 RADIAL : RAYON
155 RÉACTION
156 RRHÉE : ÉCOULEMENT
157 SAURE : LÉZARD
158 SCRIPTEUR : QUI ÉCRIT
159 SE : AFFECTION LIÉE À
160 SEXE
161 SEXUEL
162 SCAPHE : BARQUE
163 SCOPE : REGARDER
164 SOCIOLOGIE
165 SOLAIRE : SOLEIL
166 SOMATIQUE : CORPS
167 SPHERE
168 STAT : ARRÊT
169 STATIQUE
170 STHESIE : SENSIBILITÉ
171 STICHE : VERS
172 STIQUE : LIGNE
173 STOME : BOUCHE
174 STROPHE : SILLON

175 STYLE : COLONNE
176 SYNCHRASIE : AVEC
177 SYNTHESE : REUNIR
178 TAPHE : TOMBEAU
179 TAXIE : PROLONGEMENT
180 TECHNIQUE : ID
181 TÈLIE : LOIN
182 TÈRE : PORTER
183 THEISME : RELIGION
184 THELIAL : MAMELON
185 THÈQUE : ARMOIRE
186 THÈRAPEUTHE : QUI SOIGNE
187 THERME : CHALEUR
188 THÈSE : POSER
189 TOME : COUPER
190 TONE : TON
191 TOPE : LIEU
192 TRON : MACHIN
193 TIQUE
194 TROPHISME : CROISSANCE
195 TROPE : QUI TEND VERS
196 TROPISME : " "
197 TYPE : CARACTÈRE
198 UNIVERS : ID
199 URIE : URINE
200 VISION : ID
201 VORE : QUI DÉVORE
202 XYLE : BOIS
203 ZOAIRE : ANIMAL
204 ZIGOTE : ATTELÉ
205 N : TRUC



333 débuts de mots, 205 fins de mot, (333 x 205) = 68265 mots potentiels.
Le LOGOTRON complet nécessite une mémoire centrale de 32 K.

— Mon dieu, quelle LOGORRHÉE !





— Monsieur lemur !!

— Euh... oui, monsieur le directeur...

— Monsieur lemur, je vous rappele que notre chiffre des ventes baisse, que nos exportations sont menacées, et que nous cherchons ici un moyen de palier à cette crise qui nous atteint !

— Monsieur le directeur, je suis confus...

— Bon, revenons à ce programme LOGOTRON, ou NEOTRON. qu'est-ce-que cela donne ? Allons nous trouver parmi tous ces néologismes une idée de produit nouveau à fabriquer et à commercialiser

— Je l'espère, monsieur le directeur, je l'espère...

Pendant que monsieur Lemur et madame Pinardel s'absorbaient dans une fastidieuse **ENTRÉE DE DONNÉES** sur les machines de table (le logotron complet entre très bien sur un ORIC-1 ou un APPLE II, ou toute autre machine ayant une MÉMOIRE CENTRALE suffisante) Achille Moneyback se pencha sur ce qui sortait sur l'imprimante de la machine de Poche SHARP PC-1251.

– Voyons, un **BUTYROSCOPE**, cela pourrait être une machine pour observer la couleur du beurre

– Hum ....

– Ah, un **PANSCAPHE**, c'est intéressant, non. PAN veut dire TOUT, et SCAPHE : barque, véhicule.

– Autrement dit, remarqua Sophie c'est un véhicule pour aller n'importe où ...

– Oui, c'est un peu trop général, ce me semble.



Et un **LATÉROSCOPE** ?

– Cela peut être un appareil pour regarder sur les côtés. Pas très convainquant...

– Que penses-tu du **ZOODYNE** ?

– C'est une machine à traction animale. Par exemple un char à Boeufs.

– Ah ....

Le logotron tourna, tourna ....
Mais il n'apportait pas la solution.
Sophie et l'oncle Achille consignèrent quelques trouvailles :

MÉSOGRAPHE : Appareil pour écrire entre les lignes
PHALLOSTAT : Slip Kangourou
ARCHÉOSCAPHE : Machine pour voyager dans le passé
ANISOPODE : Aux pieds dissemblables : exemple DAHU.
ARCHÉOPLANE : Cerf-volant
ARSÉNOTROPE : Recherchant la virilité.
BIBLIOSTAT : Presse-livre
BRACHYCERQUE : A queue courte
COSMOPHOBE : Se dit de quelqu'un qui ne peut pas supporter l'Univers
CHRONOTHÉRAPEUTE : qui soigne en laissant faire le temps
CHRYSORCHIDE : Personne qui se fait les c..... en or.
ELASTOSAURE : Ancêtre du ver de terre.
ÉPISTÉMOZOAIRE : Animal savant
PANGAME : qui a tendance à épouser tout le monde
HÉLIORRAGIE : Eruption solaire.
HYPOLITHE : Sous-pierre
HÉMIGAME : A moitié marié
LOGOTHÉRAPIE : Psychanalyse
LOGOTOME : Pour faire des phrases hachées.
LATÉROGRADE : qui marche à côté de ses chaussures
PANMANIE : Manie de tout
MYCOCLASTE : Casseur de champignon
NÉCROGRAMME : Faire-part de décès.

ORDINOSAURE : ancêtre de l'ordinateur.
PSEUDOCRATE : Sans autorité réelle
PYROTOPE : cheminée
PORNOTROPISME : attrait pour la litterature cochonne.
PALÉOGAME : qui se reproduit de façon désuète.
PANPHOBIE : Horreur de tout.
PODOCLASTE : Casse-pied
PLOUTARQUE : Dont l'autorité s'appuie sur l'argent.
SCHIZOCRATE : qui tranche sur tout.
THÉOTOPE : Paradis
GÉRONTOMACHIE : Combat de vieux.
NÉCRODYNE : Corbillard
PACHYDROME : Déviation pour poids lourds
COSMODRAME : Big bang
ÉPISTÉMOSAURE : Vieux schnock de l'académie des sciences.
SEXON : Particule qui passe son temps à se reproduire
ÉPISTEMOLITHE : Pierre philosophale.
ORCHIDOCLASTE : Casse-couille
CRYOLITHE : glaçon
ÉLASTOLITHE : variété de pierre élastique
ÉLASTODENDRON : Arbre à chewing-gum
PANGÈNE : Situation où tout est possible.
PANLOGIE : Etude de tout.
LOGODROME : Assemblée Nationale
PALÉOTOPE : Maison de retraite
PSEUDOCLASTE : qui ne casse rien.

PSEUDON : Particule dont l'existence n'est pas prouvée.
LATÉROLOGUE : Spécialiste dans une discipline qui n'est pas la sienne.
NÉCRODROME : Piste d'entrainement pour zombies.
CHRONODROME : Terrain d'atterrissage pour chronoscaphe
CHRONOSCAPHE : Appareil pour voyager dans le Temps.
MONOSCOPE : Petit cylindre en matériau quelconque, permettant de ne regarder qu'une seule chose à la fois.
BACTÉRIOMANCIE : Prédiction de l'avenir en regardant dans un microscope.
CRYOTROPISME : afflux saisonnier dans les stations de sports d'hiver
STATODYNAMIQUE : Etude de l'évolution des états stationnaires.
CHRONULE : animal vivant dans l'hyperespace.
PALÉOMANIE : Goût des antiquités.
LOGOMÈTRE : appareil à mesurer les discours.

VOS TROUVAILLES PERSONNELLES :

Fouchtre, nous avons créé des centaines et des centaines de mots, et je ne vois pas poindre quelquechose qui puisse convenir...

PRRRRÎT !
PRRRRiiiT !

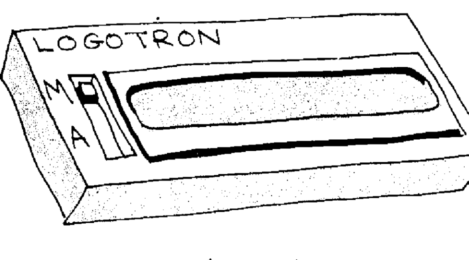
monsieur le directeur, si je puis me permettre... n'avons-nous pas la réponse sous les yeux ?

— Parlez, monsieur lemur, parlez !
— Eh bien, est-ce que le **LOGOTRON** lui-même ne constitue pas, pourrait-on dire, une machine à créer du langage.
— Une ?... mais sapristi, lemur, vous avez raison ! Ce qu'il faut fabriquer, ce sont des Logotrons, tout simplement ! Que faut-il ? trois fois rien. Un affichage à cristaux liquides, un microprocesseur, un peu de mémoire...

Il faut bien convenir que, ce qui se vend le mieux, de nos jours, ce sont les mots !

Et il est clair que ce que le LOGOTRON nous a révélé, c'est que nous vivions à la surface d'un gisement sémantique quasiment inexploité !

– le monde moderne a besoin de mots nouveaux. Et nous allons les lui donner. On peut envisager des LOGOTRONS spécialisés, dans différents domaines. (*)
– Monsieur le Directeur, je vous ferai remarquer qu'en la matière le corps médical a déjà une sérieuse avance...
– Qu'importe. Désormais, les gens ne chercheront plus leurs mots. Vive la logotronique !



Madame Pinardel intervint :
– Ces mots nouveaux ne seront pas utilisables si nous ne fournissons pas en même temps leur signification, leur "mode d'emploi". Tout le monde ne connait pas le grec et le latin...
– C'est vrai, comment allons-nous faire ?

(*) L'auteur est toujours à la recherche d'un industriel prêt à tenter l'aventure.

LE LOGOTRON AVEC SENS DES MOTS

— C'est très simple, dit Sophie, nous allons utiliser quatre fichiers au lieu de deux. D$(I) contiendra les débuts de mots, F$(J) les fins de mots, P$(I) la signification des débuts de mots et Q$(J) les significations des fins de mots.

— Ça va être compliqué, dit Moneyback. Il y aura quatre paquets de data, et quatre séquences de chargement, de remplissage des fichiers. Il va falloir réserver de la place mémoire pour ces fichiers P$(I) et Q$(J). Le plus simple est de donner aux chaînes P$(I) et Q$(J) la même longueur de 11 caractères. Et il faudra faire très attention pour qu'il y ait bien correspondance entre les racines verbales et leur signification.

il y a une solution très simple et très sûre pour éviter de se tromper

laquelle ?

— Nous allons constituer **SIMULTANÉMENT** les fichiers D$(I) et P$(K). Puis, de la même manière les fichiers F$(J) et Q$(L). Et cerise écrivit :

```
1 REM LOGOTRON DATA+SENS
2 DIM D$(38)*11 : DIM P$(38)*11 : DIM F$(38)*11 : DIM Q$(38)*11
1Ø DIM M$(Ø)*24 : DIM S$(Ø)*24 : I = -1
2Ø I = I+1
3Ø READ D$(I), P$(I)
4Ø IF D$(I)="*" THEN 6Ø
5Ø GOTO 2Ø
6Ø D = I-1
7Ø J = -1
8Ø J = J+1
9Ø READ F$(J), Q$(J)
1ØØ IF F$(J)="*" THEN 12Ø
11Ø GOTO 8Ø
12Ø F = J-1
13Ø RANDOM
14Ø I = RNDD : J = RNDF
15Ø M$(Ø) = D$(I) + F$(J)
16Ø S$(Ø) = Q$(J)+"-"+P$(I)
17Ø WAIT 1ØØ : PRINT M$(Ø)
18Ø PRINT S$(Ø)
19Ø GOTO 14Ø
```

Pour les ordinateurs de table (ORIC-1, TO7, etc...)

```
2 DIM D$(3Ø) : DIM F$(3Ø) :
DIM P$(3Ø) : DIM Q$(3Ø)
```

et...

```
15Ø M$ = D$(I) + F$(J)
16Ø S$ = Q$(J) +"-"+P$(I)
17Ø WAIT1ØØ : PRINTM$
18Ø WAIT1ØØ:PRINT S$
19Ø GOTO 14Ø
```

Suivent les DATA : (les guillemets sont inutiles sur ORIC, TO7, etc..

```
200 DATA"CLEPTO","VOL","CHRONO","TEMPS","EPISTEMO","SCIENCE",
"LOGO","DISCOURS",
210DATA "PAN","TOUT","PATHO","MALADIE","HEMI","A MOITIE",
"CRYPTO","CACHE"
220 DATA"EROTICO","EROTISME","GNOSEO","CONNAISSANCE",
"ORCHIDO","COUILLE","PARALLELO"
230 DATA"PARALLELE","SEMIO","SENS","ERGO","TRAVAIL",
"GERONTO","VIELLARD"
240DATA "NECRO","MORT","MESO","MILIEU","PODO","PIED",
"PORNO","PORNO"
250DATA "PROTO","PREMIER","SCATO","EXCREMENT","DOXO",
"OPINION","PLOUTO"
260 DATA "RICHESSE","PHONO","SON","INFLATO","GONFLEMENT"
"PYRO","FEU"
270DATA"RETRO","EN ARRIERE","PSYCHO","AME","LITHO","PIERRE",
"MACRO","GRAND"
280 DATA"CHRYSO","OR","PHILO","AMOUR","MICRO","PETIT",
"TECHNO","TECHNIQUE","BIO"
290 DATA"VIE","SPELEO","CAVERNE","TOMO","COUPER","TOPO",
"LIEU"
299DATA"*","*"
```

```
300 DATA "GAME","MARIAGE","SE","AFFECTION","EDRE","EDIFICE",
"DYNE","ENERGIE"
310 DATA "CERQUE","QUEUE","DROME","ROUTE","PHOBE","QUI DETESTE",
"PATHE","
320 DATA "MALADE DE","SCAPHE","VEHICULE","TOPE","LIEU"
"TRON","MACHIN"
330 DATA "N","PARTICULE","ME","TUMEUR","METRE","MESURE","NAUTE"
340 DATA "QUI NAVIGUE","DRAME","THEATRE","IDE","PRESQUE","COSME",
"UNIVERS DE"
350 DATA "SOME","CORPS","SPHERE","SPHERE DE","STASE","ARRET",
"TAPHE","TOMBEAU"
360 DATA "TROPE","TENDANCE","POLITIQUE","MANE","MANIAQUE DE",
"GRADE"
370 DATA "MARCHER","GRAMME","MESSAGE","GENE","GENERATEUR",
"ELASTIQUE","ELASTIQUE"
380 DATA "CINESE","MOUVEMENT","CRATE","POUVOIR","DYNAMI
QUE","DYNAMIQUE","CYTE"
390 DATA "CELLULE"
399 DATA "*","*"
```

– La ligne importante, dans ce programme, est évidemment :

```
30 READ D$(I),P$(I)
```

Comme tu peux le voir elle comporte une **VIRGULE**. Et elle signifie : prendre les deux données qui suivent et les mettre, la première dans D$(I) et la seconde dans P$(I).

Moneyback soupira :

— La façon dont sont lues ces données présentes dans ces lignes de **DATA** me semble un peu mystérieuse, non ?

— Mais non, c'est tout simple. Lorsque l'ordre **READ** est rencontré, dans l'exécution du programme, pour la première fois, l'ordinateur cherche la première ligne portant le repère, l'instruction **DATA**. Un POINTEUR DE DATA se trouve alors positionné sur la première donnée (en l'occurrence la chaîne de caractères CLEPTO. Puis cette donnée est stockée dans le fichier qui suit l'ordre READ. Ce sera donc D$(I) avec I = Ø. La virgule entraîne un mouvement vers le bas du pointeur, qui se positionne devant "VOL", qui est alors stocké dans D$(I) avec I = Ø. Et ainsi de suite, jusqu'à lecture complète des DATA. La seule précaution à prendre est de ne pas demander à l'ordinateur de lire plus de DATA qu'il n'en possède dans son programme. Sinon, il réagira par un message d'erreur.

```
18Ø WAIT1ØØ : PRINT S$(Ø)
19Ø GOTO 14Ø
2ØØ DATA
"CLEPTO",
"VOL",
"EPISTEMO",
"SCIENCE",
"LOGO",      <— POINTEUR DE DATA
"DISCOURS",
21Ø DATA
"PAN",
"TOUT",
"PATHO",
"MALADIE",
"HEMI",
"A MOITIE",
"CRYPTO",
"CACHE",
22Ø DATA
"EROTICO",
```

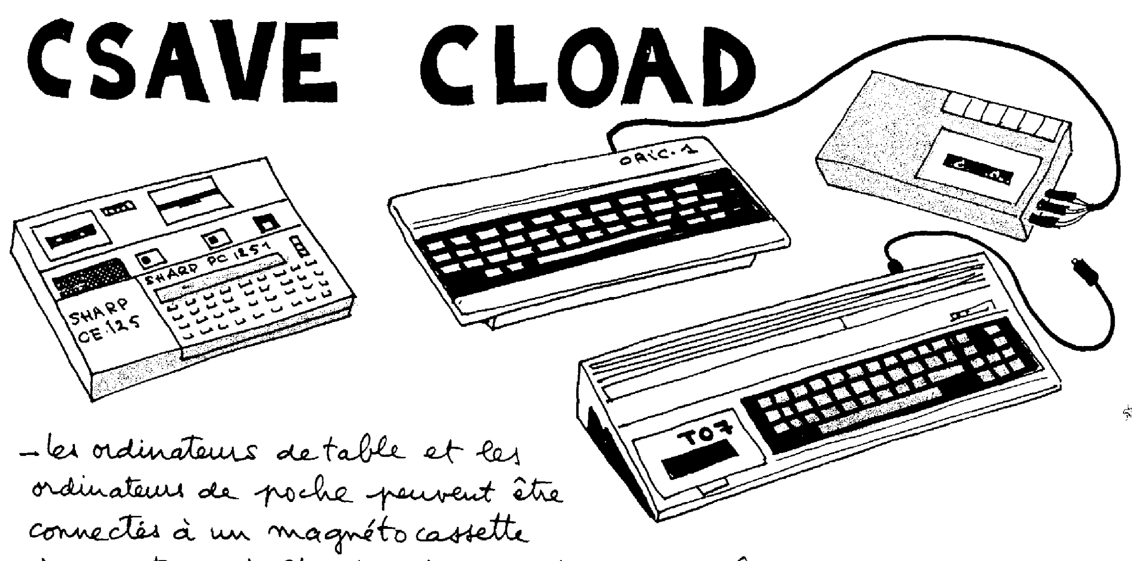
Hmm...

Au rayon antiquités....

– Alors on peut constituer deux fichiers avec l'instruction READ?
– Tu peux en constituer trois, quatre, etc... à condition de séparer les fichiers par une virgule :

READ D$(I), P$(I), F$(I), Q$(I)

CSAVE CLOAD



– les ordinateurs de table et les ordinateurs de poche peuvent être connectés à un magnétocassette à compteur, à l'aide d'un cordon spécial à trois fiches(*)
Cet appareil est utilisé comme **MÉMOIRE EXTERNE.**
Et l'information est mémorisée sur la bande magnétique.
Lorsqu'on veut transférer le contenu de la mémoire centrale dans la mémoire externe, on commence par positionner le magnétophone sur une portion de bande vierge, en s'aidant du compteur. Puis on compose au clavier **CSAVE** suivi du nom du programme, entre guillemets. Exemple CSAVE"PROGR.4"

(*) le **TO7** a une prise et un magnétophone spécial.

le transfert va s'effectuer à l'aide du fil habituellement relié à un microphone. Il sera, bien sûr, nécessaire d'enfoncer préalablement la, ou les touches d'ENREGISTREMENT du magnétophone. La broche centrale permet de déclencher le mouvement de la bande directement à partir du clavier de l'ordinateur, et celui-ci démarrera dès que la touche [Fin de message], suivant l'ordre CSAVE "PRO-1" aura été pressée. Il s'arrêtera automatiquement.

Le nombre de caractères, entre guillemets, permettant de repérer le programme, dépend de la machine utilisée. Sur le PC 1251 ce nombre est SEPT.

L'opération inverse est le **CHARGEMENT**. C'est à dire la recopie du contenu de la bande dans la mémoire centrale de l'ordinateur. Après positionnement correct de la bande, on presse la touche PLAY. Puis on compose **CLOAD**, suivi des caractères de repérage du programme.

– Bon, dit Moneyback, je vais appeler ce programme "LOGO 1".

LE PROGRAMME CATALOG

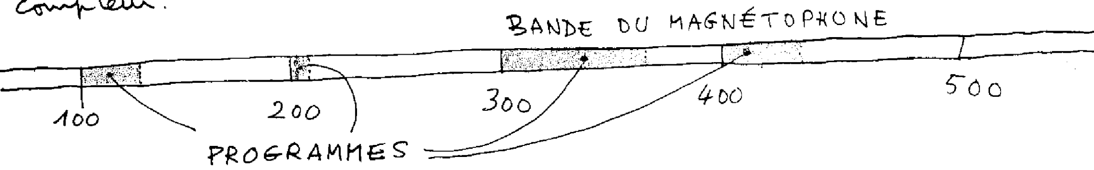
— Tu vas écrire des programmes, et puis, très vite, tu ne te souviendras plus, ni de leur nom, ni de leur emplacement sur la bande... Et quand des mois auront passé, tu ne te souviendras plus de ce que ces programmes signifiaient.

— On peut mettre des **REM** dans les programmes.

— Bien sûr. Et tu n'as pas nécessairement besoin de connaître le nom d'un programme pour pouvoir le charger. Sur le PC 1251, tu peux composer simplement CLOAD, et la machine chargera le premier programme rencontré sur la bande. Sur l'ORIC-1 l'ordre équivalent est CLOAD"". Mais, imagine que tu aies perdu tes notes au sujet d'une cassette. Il te faudra charger un à un tous les programmes qu'elle contient, puis essayer de t'y retrouver à l'aide des REM et des LISTING de ces programmes.

— Laborieux...

— Par précaution, je te propose un certain nombre de mesures sages. Primo, toujours débuter les sauvegardes avec des numéros de compteur faciles à repérer : Ø, 1ØØ, 2ØØ, 3ØØ (sur le compteur). Tes programmes n'atteindront pas, en général, les cent unités du compteur.



- Une bonne solution consiste à inscrire, en début de bande, un programme CATALOG, qui contient tous les renseignements utiles : cote des programmes sur la bande, longueur, contenu.

```
1 REM"PROGRAMME CATALOG"
2 REM" D(I) = ADRESSE DEBUT SUR BANDE"
3 REM"F(I) = ADRESSE FIN SUR BANDE:N$(I)=NOM PROGRAMME"
4 REM"R$(I)=RENSEIGNEMENTS DIVERS,LIMITÉS À 72 CARACTÈRES"
5 DIM D(10):DIM F(10):DIM R$(10)*72 :DIM S$(0)*24
7 D(0)=10 : F(0)=30 : N$(0)="CATALOG"
8 R$(0)="ADRESSE/BANDE; NOM PROG RENSEIGNEMENTS DIVERS"
10 D(1)=100 : F(1)=140 : N$(1)="LOGO1"
11 R$(1)="LOGOTRON AVEC DATA"
20 D(2)=200 : F(2)= 240 : N$(2)="LOGO2"
21 R$(2)="LOGOTRON AVEC DATA PLUS SENS DES RACINES
     PREFIXES ET SUFFIXES"
30 D(3)=300 :F(3)=360 : N$(3)="LOGO3"
31 R$(3)="LOGOTRON AVEC CHARGEMENT FICHIER
199 I=-1 : REM INITIALISATION INDICE PROGRAMME
200 I=I+1 :REM INCREMENTATION
205 IF N$(I)=""THEN END :REM SORTIE DE BOUCLE
210 WAIT 100 : PRINT"DEBUT EN ";D(I)
220 WAIT 100 : PRINT"FIN EN ";F(I)
230 WAIT 100 : PRINT"NOM : ";N$(I)
```

On calcule ensuite la longueur L de la chaîne R$(I) :

```
240 L=LEN(R$(I))
```

La chaine R$(I) peut avoir une longueur quelconque, inférieure à 73. Nous allons afficher les 24 premiers caractères :

```
25Ø S$(Ø)=LEFT$(R$(Ø),24):REM PREMIERE SOUS-CHAINE
26Ø WAIT 1ØØ : PRINT S$(Ø)
```

Si la chaine a une longueur inférieure à 25 il faut passer au programme suivant :

```
27Ø IF L<25 THEN 2ØØ
```

On affiche alors les caractères 25 à 48 :

```
28Ø S$(Ø)=MID$(R$( ),25,48):REM SECONDE SOUS-CHAINE
29Ø WAIT 1ØØ : PRINT S$(Ø)
3ØØ IFL<49 THEN 2ØØ
31Ø S$(Ø)=MID$(R$(I),49,L):REM TROISIEME SOUS-CHAINE
32Ø WAIT 1ØØ : PRINT S$(Ø)
34Ø GOTO 2ØØ
```

La sortie de boucle est assurée à la ligne 2Ø5. En effet, la boucle réalise une exploration des fichier, et en particulier du fichier N$(I). Dès que l'élement de ce fichier ne contient rien, c'est à dire le caractère vide "", alors ordre est donné de s'arrêter (END).

A chaque fois qu'on stockera un nouveau programme sur bande, il faudra modifier ce programme catalog, en conséquence. Et lui-même sera stocké à partir de la cote 1Ø de la bande magnétique (et non zéro, pour éviter l'amorce)

Moneyback lança ce programme CATALOG et vit apparaître successivement :

DEBUT EN 10
FIN EN 30
NOM : CATALOG
ADRESSES/BANDE NOM PROG
RENSEIGNEMENTS DIVERS
DEBUT EN 100
FIN EN 140
NOM : LOGO 1
LOGOTRON AVEC DATA
DEBUT EN 200
FIN EN 240
NOM : LOGO 2
LOGOTRON AVEC DATA PLUS
SENS DES RACINES
PREFIXES ET SUFFIXES

etc... etc...

L'informatique, ça me plaît mieux maintenant que je commence à y comprendre quelque chose

– Bon, dit Moneyback. Nous avons donc un LOGOTRON avec DATA, qui donne une indication sémantique sommaire, c'est à dire qui traduit les racines utilisées. C'est bien. Mais que reste-t-il ? pas grand chose. Trente huit débuts de mots et trente huit fins de mots. Soit, potentiellement mille quatre cent quarante quatre mots.

PRINT# INPUT#

— Oui, dit monsieur Lemur, il est clair que ce système des DATA implique une **DOUBLE ÉCRITURE EN MÉMOIRE**, une fois dans le programme, et une fois dans les registres de données.
— Le programme lui-même, celui qui crée les néologismes, est ridiculement court :

```
130 RANDOM
140 I=RND D : J=RND F
150 M$(0) = D$(I) + F$(J)
160 S$(0) = Q$(J)+"-"+ P$(I)
170 WAIT 100 : PRINT M$(0)
180 WAIT 100 : PRINT S$(0)
190 GOTO 140
```

(écriture PC 1251)

Sans oublier `2 DIM D$(70)*11 : DIM F$(70)*11 : DIM P$(70)*11 : DIM Q$(70)*`
`DIM M$(0)*24 : DIM S$(0)*24`

Ce qui fait 161 pas de programme. Et il en reste 3325 !!!
— On devrait pouvoir les utiliser de manière plus économique.

Sophie apporta la solution

— Il faut utiliser des instructions **INPUT#** et **PRINT#** qui permettent de transférer des blocs de données.
— Comment procède-t-on ?
— Supposons que nous ayions "à la main" rentré nos données dans les registres. Il s'agit de D et F, effectifs des ensembles

des débuts de mots et fins de mots, puis des fichiers D$(I) et F$(J), et des "traductions" P$(I) et Q$(J).
Le transfert sur bande s'effectuera (après avoir mis le magnétocassette en configuration d'enregistrement) en composant

PRINT # "FICHIER"; D, F, D$(*), F$(*), P$(*), Q$(*)

L'ordinateur recopiera sur bande ces données. C'est à dire les contenus de D, de F, et les fichiers D$(I), F$(J)., P$(I), Q$(J)

— Je suppose que l'ordre :

INPUT # "FICHIER"; D, F, D$(*), F$(*), P$(*), Q$(*)

va réaliser l'opération inverse?
— Oui, à condition que tu aies prévu les emplacements mémoire pour les fichiers à l'aide d'ordres DIM.
— Et la chaine entre guillemets sert d'identificateur, de repère.
— Oui, et on a droit à 7 caractères.
— autrement dit, le programme se résumerait à :

```
1 REM "LOGOTRON + CHARGEMENT PAR INPUT #"
2 DIM D$(70)*11 : DIM F$(70)*11 : DIM P$(40)*11 : DIM Q$(70)*11
DIM M$(0)*24 : DIM S$(0)*24
5 INPUT # "RACINES"; D, F, D$(*), F$(*), P$(*), Q$(*)
10 RANDOM
20 I = RND D : J = RND F
30 M$(0) = D$(I) + F$(J) : WAIT 100 : PRINT M$(0)
40 S$(0) = Q$(J) + "-" + P$(I) : WAIT 100 : PRINT S$(0)
50 GOTO 20
```

SAiSiE

— Ce programme permet le transfert de 28Ø données. Soit 7Ø débuts de mots, 7Ø fins de mots, et leurs significations. 4900 mots potentiels. Dans cette version c'est la capacité maximale du PC1251. Si on ne donne pas les traductions des racines verbales, on passe à 140×140 = 19600 mots potentiels.

— Reste l'opération fastidieuse qui consiste à rentrer toutes ces données en mémoire...

Bon, plus qu'à rentrer tout cela...

MÉMOIRE
EXTERNE
À DISQUE
10 MEGA OCTETS

Très souvent, ceux qui sont soudain saisis par la Grande Fièvre Informatique ne se rendent absolument pas compte du volume de travail que représente l'entrée des données

!
ooo



PROGRAMME SAISIE

– Dans la **SAISIE DE DONNÉES** il faut se débrouiller pour n'avoir à taper que le STRICT MINIMUM, et il est tout à fait indiqué d'utiliser un PROGRAMME DE SAISIE, comme par exemple :

```
1 REM"PROGRAMME DE SAISIE"
5 DIM D$(7Ø)*11:DIM F$(7Ø)*11:DIM P$(7Ø)*11:
DIM Q$(7Ø)*11:DIM M$(Ø)*24:DIM S$(Ø)
*24
10 FOR I=Ø TO 7Ø
2Ø INPUT"PREFIXE :";D$(I)
3Ø INPUT"SIGN.";P$(I)
4Ø NEXT I
5Ø PAUSE "SUFFIXES"
6Ø FOR J=Ø TO 7Ø
7Ø INPUT"SUFFIXE";F$(J)
8Ø INPUT"SIGN.";Q$(J)
9Ø NEXT J
1ØØ PAUSE "FIN DE SAISIE"
11Ø PRINT# "RACINES";D,F,D$(*),F$(*),
P$(*),Q$(*)
```

Grâce à ce programme, Moneyback put rentrer rapidement en mémoire centrale ses débuts et fins de mots, la dernière ligne concrétisant le stockage sur la bande. Il s'arrangea pour que ce fichier suive le programme de la page 69, également inscrit sur cette mémoire externe. Et le tout composa un LOGOTRON tout à fait convenable.

ÉPILOGUE

Quelques mois plus tard...



— Tu sais comment sont ces gens. Ils adorent faire des mots très longs, avec plusieurs racines. Le LOGOTRON était fait pour eux.
Nos LOGOTRONS "parlent" déjà dix sept langues, y compris le bantou et le Tamoul. Les chefs d'état sont de bons clients.
— Il est vrai que leur rôle se résume souvent à produire du langage.
— Nous avons des logotrons par SPÉCIALITÉS. Des choix ad hoc de racines verbales conduisent à des néologismes se référant,

sélectivement, au monde médical, à la technologie, à l'économie...
On peut aussi combiner un début de mot que l'on choisi,
avec des fins de mots tirés au hasard. Ou l'inverse (*)



(*) Modifiez votre programme LOGOTRON pour avoir ce résultat.

Parce que les VOITURES FRANÇAISES sont les SEULES sur le marché international, a être équipées de :

L'INSULTOPHONE

```
1 RANDOM
2 REM"PROGRAMME AD HOC"
10 DIMD$(70)*11:DIMF$(70)*11
20 PAUSE"UNE MINUTE"
30 WAIT 150
40 PRINT"JE CHARGE MES DONNÉES"
50 I=-1 :REM"INITIALISATION"
60 I=I+1:REM"INCRÉMENTATION"
70 READ D$(I)
80 IF D$(I)="*"THEN100
90 GOTO 60
100 D=I-1
110 J=-1 :REM"INITIALISATION"
120 J=J+1:REM"INCRÉMENTATION"
130 READ F$(J)
140 IF F$(J)="*" THEN 130
150 GOTO 120
130 F=J-1
```

```
140 WAIT 100 : PRINT "FIN DE CHARGEMENT"
150 M$ = INKEY$
```

Dans le registre M$ se trouve le dernier caractère pressé au clavier. Ceci permet de donner des ordres à l'ordinateur, en continu, sans stopper le cours du programme

```
160 IF M$ = " " THEN 190
170 WAIT 80 : PRINT "PRET POUR BORDEE"
180 GOTO 150
190 M$ = "" : REM "REMISE CARACTÈRE VIDE DANS M$"
200 WAIT 80 : PRINT "VA DONC, EH !"
210 I = RND D : J = RND F : REM CHOIX ALEATOIRE I ET J
220 [ACTIVATION SYNTHÉTISEUR PAROLE]
230 REM CONTROLE ECRAN
240 WAIT 100 : PRINT " "; D$(I) ; F$(J) : REM "AFFICH. NEOLOGISME"
250 M$ = INKEY$ : REM "SAISIE CARACTERE CLAVIER"
260 IF M$ = " " THEN 140
270 GOTO 210
300 DATA "EPISTEMO", "PAN", "PATHO", "HEMI"
310 DATA "ELASTO", "EROTICO", "GNOSEO", "ORCHIDO", "PARALLELO"
320 DATA "GERONTO", "NECRO", "ARCHEO", "MYCO", "PODO", "PORNO"
"PROTO", "SCATO", "COPRO"
330 DATA "DOXO", "PLOUTO", "INFLATO", "THEO", "ZOO", "SEXO", "PYRO"
"RETRO"
```

```
340 DATA "PSYCHO","LITHO","MACRO","CHRYSO","PHILO","MICRO","TECHNO",
"VIDEO"
350 DATA "PAPYRO","SPELEO","RHINO","PHYTO","PHOTO",
360 DATA "HETERO","CEPHALO","CRYO","TERATO","THERMO","PHRENO","*"
370 DATA "GAME","CERQUE","PHOBE","PATHE","PHRENE","PTERE"
380 DATA "ZOAIRE","NAUTE","BLASTE","GASTRE","PETE","PHILE"
390 DATA "SAURE","LOGUE","MORPHE","PHORE","PLASTE","TROPE"
400 DATA "MANE","ZIGOTE","GRADE","CRATE","CYTE","SEXUEL"
410 DATA "PHRENE","PHAGE","CEPHALE","THERAPEUTE,"PHILOSOPHE"
420 DATA "CHRONE","DIDACTE","LATRE","PITHEQUE","*"
```

→ Ces choix de racines permettent de composer des injures tout à fait acceptables au nombre de 1496 !



Après une séance classique (lignes 1 à 13ø) d'arrangement des DATA dans les fichiers, l'ordinateur se trouve mis dans une boucle d'attente. L'instruction **INKEY$** indique quelle est la dernière touche qui vient d'être frappée.
Si on frappe sur la barre d'espacement (caractère espace SPC sur le PC1251) on déclenchera la sortie de boucle.

A la ligne 19ø on s'empresse de "remettre à zéro" M$ en y inscrivant le "caractère vide" (à ne pas confondre avec le caractère espacement " ").
Ligne 24ø : la saisie directe d'un caractère au clavier (espace " ") permet d'interrompre la bordée d'injure.

Un synthétiseur de parole est au départ un synthétiseur de fréquences sonores, qui, composées, donnent des PHONÈMES, des éléments de langage parlé. Les mots sont des compositions de phonèmes. L'Insultophone sera un microordinateur où on aura mémorisé les phonèmes de base, dans la langue choisie.

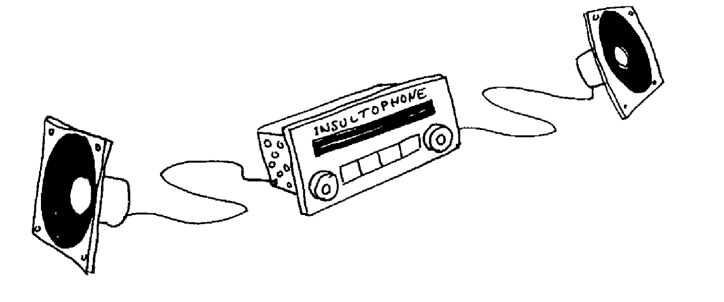
Et maintenant, quels sont tes projets ?

Une nouvelle machine, basée sur l'informatique, et qui fera l'objet du prochain livre. Mais... chut... secret !

FIN

L'Histoire qui vous a été contée est imaginaire. Néanmoins l'**INSULTOPHONE**, invention qui sort l'entreprise d'Achille Moneyback du marasme, pourrait très bien être construit. Invention cent pour cent française, adaptable dans toutes les langues et qui devrait équiper tout véhicule de série, au même titre que le rétroviseur, la ceinture de sécurité ou l'allume-cigare.

Tout industriel éventuellement intéressé peut entrer en relation avec l'auteur en écrivant à la maison d'édition. Nombreuses inventions disponibles.



Jean Pierre Petit